Treball de Fi de Màster

Màster en Enginyeria de l'Energia

Numerical Simulation of the Navier-Stokes Equations using Finite Volume Method

MEMÒRIA

Autor: LIANG, SHANGWU
Director: CONSUL SERRACANTA, RICARD
Codirector: RUIZ MANSILLA, RAFAEL

Convocatòria: Junio 2017

Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona

Resume

This project is to develop a finite volume code to solve the Navier-Stokes (NS) equations coupled with the energy equation in two dimensional Cartesian coordinates. The codes thus developed are verified and can be directly used to analyze various fluid mechanics and heat transfer phenomena.

Before the final code, there has been a process from the introduction to numerical simulation to solving some basic problems, such as diffusion equation, convection-diffusion equation, lid driven cavity problem. The results of the code have been verified by analytical solutions or benchmarks. And finally, the differential heated cavity problem has been solved.

For lid driven cavity case and the differential heated cavity case, they are also simulated using a commercial computational fluid dynamics (CFD) software, ANSYS Fluent. The results thus obtained from the code and ANSYS are compared with the benchmark solutions for the two cases available in published journals. A comparative study of these results has been presented in this project.

Summary

1. Introduction

To fully investigate a physical process, experimental investigation can be done with actual measurement. But to do a full-scale test is often too expensive, and there are serious difficulties of measurement in many situations. The alternative then is to perform a theoretical calculation.

The analysis of fluid flow, heat transfer and other related can be described by a few governing equations. In such cases, the governing equations to describe the phenomenon are complex differential equations consisting of a number of interdependent variables. It is extremely difficult to obtain analytical solution, especially with complicated geometries. Computational fluid dynamics (CFD) deals with these kinds of problems. With a constructed mathematical model from those differential equations, we can solve those equations to approximating final results iteratively by a computer.

In this project, several C language programs are developed for solving some of the most common governing equations that appear in most of fluid flow and heat transfer problems. For each case, grid refinement studies have been performed in order to assess that the code is free of numerical errors, and the solutions trend to a result similar to the referenced one.

1.1. Objective of the project

The objective of the project is to develop a profound understanding of numerical simulation by programming its own code for several typical problems, and then compare the results with analytical solutions, benchmarks and commercial CFD software. Meanwhile, a deeper understanding of fluid flow and heat transfer has also been formed.

1.2. Scope of the project

Develop code with C programming language to solve diffusion equation, convection-diffusion equation, Navier-Stocks equation and Navier-Stocks equation with energy equation, Comparison analyses are conducted between the results of developed codes and analytical solution or benchmarks.

Simulate the same cases to solve Navier-Stocks equation and Navier-Stocks equation with energy equation with ANSYS Fluent. Comparison analyses are conducted between the results of developed codes and the that of the software as well.

2. Background of numerical simulation

The essence of numerical heat transfer is to use a set of units to represent the continues field and the physical variables under investigation. With the governing differential equation and boundary conditions, we can construct algebraic equation that represent relations among those units. Then with the powerful calculation ability of modern computer, an approximate solution can be obtained.

The general algorithm to solve such problem is shown in the diagram below:

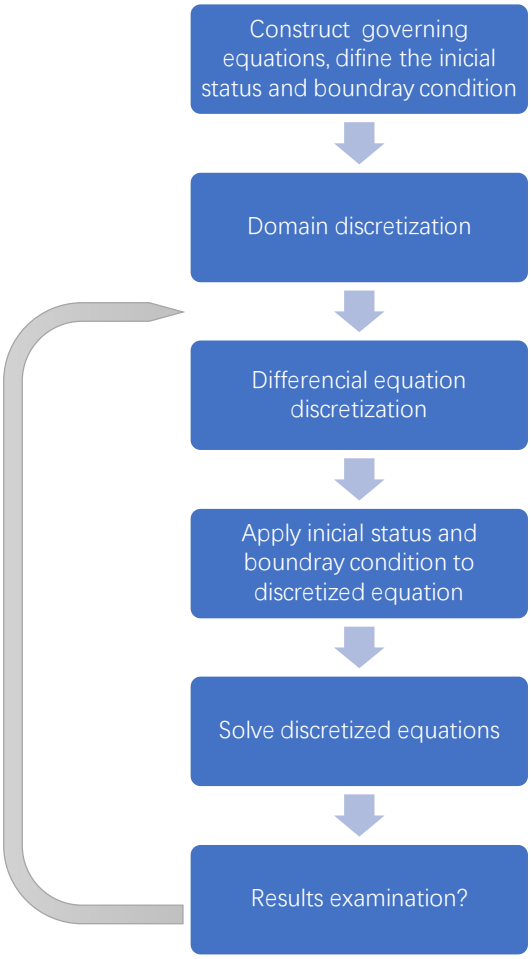


Fig. 2.1. General process to solve a physic problem by numerical method

2.1. Governing differential equations

All fluid flow and heat exchange process are governed by three basic physical rules, the mass conservation, momentum conservation and energy conservation. The first step to find the

numerical solution of fluid flow and heat transfer is to express these their conservation equations mathematically. In this project, most problems studied are two-dimension problems, there governing equation can be generally expressed as follow:

Mass conservation equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0 \qquad \textit{Ec 2.1}$$

X-Momentum equation:

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u u)}{\partial x} + \frac{\partial (\rho v u)}{\partial y} = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) - \frac{\partial p}{\partial x} + S_u \qquad \textit{Ec 2.2}$$

Y-Momentum equation:

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho u v)}{\partial x} + \frac{\partial (\rho v v)}{\partial y} = \frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial y}\right) - \frac{\partial p}{\partial x} + S_u \qquad \textit{Ec 2.3}$$

Energy equation:

$$\frac{\partial (\rho c_p T)}{\partial t} + \frac{\partial (\rho c_p u T)}{\partial x} + \frac{\partial (\rho c_p v T)}{\partial y} = \frac{\partial}{\partial x}\left(\kappa \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa \frac{\partial T}{\partial y}\right) + S_u \qquad \textit{Ec 2.4}$$

Where μ is the viscosity, t is the time, ρ is the density, S_u is the source term, p is the pressure, v is the vertical velocity component, u is the horizontal velocity component, κ is the diffusion coefficient, c_p is the heat capacity, T is the temperature.

2.2. Discretization methods

2.2.1. Domain discretization

Various methods have neem developed and can be employed to solve such problem, such as finite difference method, finite element method, finite analytic method, and finite volume method. Among them, finite volume method is a relatively popular one. It divides the domain into a set of control volume, every control volume is represented by a node. The original continuing space is therefore discretized into finite volumes.

2.2.2. Differential equation discretization

After the mesh is formed, the differential equations need to be discretized as well to write all those equations for every single node. Several methods can be employed here, such as Taylor-Series formulation and control-volume formulation. The first one emphasis more to derive the equation in a mathematic perspective. While control volume method is more likely to be understood as a physical interpretation. Control-volume formulation is used in this project.

2.3. Approximation scheme

When a differential equation is being discretized, certain assumption need to be made about the profile along every single control volume. We can assume that the value at a grid point prevails over the control volume, or we can also do a linear interpolation (also called central-difference scheme) between the grid points. The linear interpolation method is commonly used for the diffusion term. As for the convection term, since it contains the velocity, which is a vector, linear interpolation is far less accurate in some conditions. This means using this method for convection term would limit the usage of the solver, and might generate unreliable results. Thus, other scheme need to be employed here.

One popular one is called upwind scheme [3]. In this scheme, the formulation of the diffusion term is left unchanged, but the convection term is calculated as stated: the value of ϕ at an interface is equal to the value of ϕ at the grid point on the upwind side of the face. This means that ϕ is always equal to the node the fluid come from.

An improvement of upwind scheme was developed by Spalding([1], Hybrid scheme) and Patankar([2], power-law scheme), They are better approximation to the exact curve. Although somewhat more complicated, but still rather easy to compute.

2.4. Boundary condition

Typically, three kinds of boundary conditions are encountered in heat conduction. These are:

1. Given boundary temperature
2. Given boundary heat flux
3. Boundary heat flux specified via a heat transfer coefficient and the temperature of the surrounding fluid.

No matter how they are given, we can always construct an additional equation at the boundary, so that the linear algebraic equations can be solved.

2.5. Solution of the linear algebraic equation

Two basic methods are employed in this project to solve the linear algebraic equations. They are TDMA (Tridiagonal-Matrix Algorithm). And the Gauss-Seidel method.

The detail of how these two methods employed in a numerical heat transfer problem can be found in [3] .

Gauss-Seidel method is a relatively slow iterative method, in this project TDMA method would be used.

For a 2-dimensional problem, TDMA would be used first for every row of nodes, and then it would be used for every column of nodes. This whole process is named as SOR.

2.6. CFD verification and validation

As CFD is widely used by engineers and technicians for different equipment, more and more new codes have been developed with different numerical method. In this case, the accuracy and credibility of these codes need to be defined. It can be defined by the process of validation and verification.

Verification assessment determines if the programming and computational implementation of the conceptual model is correct and the code itself is well written (free of bugs). It examines the mathematics in the models through comparison to exact analytical results or benchmarks.

Validation assessment determines if the computational simulation agrees with physical reality. It compares the results predicted by the simulation with experimental data.

In this project, only verification processes are completed for several cases studied. The simulation results can be different from the exact analytical solution due to the computational errors. The main source of computational errors is in the discretization part of the numerical method. To quantify these errors, a post-processing tool have been proposed in [11] for the computational error analysis.

3. Diffusion equation

To get started with numerical simulation, a simple conduction problem is proposed. In the heat transfer point of view, this problem can be easily handled mathematically and it can be found in most of the governing equations in fluid flow problems

3.1. Governing equation

The governing equation for 2-D steady state diffusion equation for a general variable ϕ is shown below:

$$\frac{\partial}{\partial x}\left(\tau\frac{\partial \phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\tau\frac{\partial \phi}{\partial y}\right)+S=0 \qquad \textit{Ec 3.1}$$

which in this case, ϕ is the temperature, τ is the thermal conductivity, and *S* is the source term.

3.2. Domain discretization

As shown in the figure below, domain is divided into numbers of control volumes represented by each node. P represents the whole control volume in shadow. N, E, W and S are the neighboring nodes that represent their own control volume respectively. n, e, w and s represent the control volume faces for the internal node P.

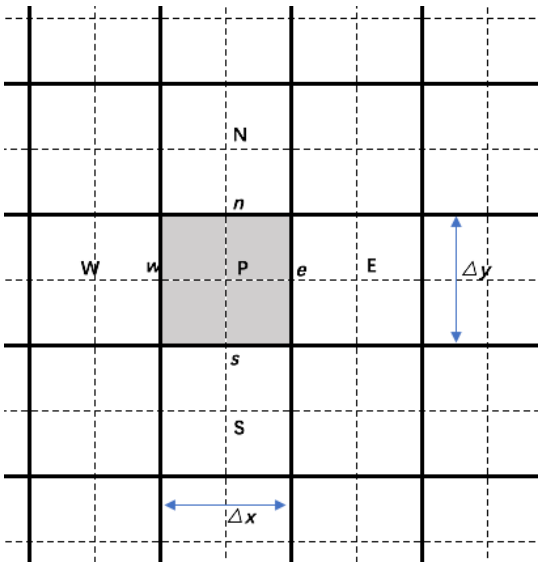


Fig. 3.1. Two Dimensional Grid used for Domain Discretization

The governing equation is intagreted in the control volume:

$$\int_w^e \frac{\partial}{\partial x}\left(\tau \frac{\partial \phi}{\partial x}\right) dx\, dy + \int_s^n \frac{\partial}{\partial y}\left(\tau \frac{\partial \phi}{\partial y}\right) dy\, dx + \int_{\Delta V} S_\phi\, dx\, dy = 0 \qquad \textit{Ec 3.2}$$

After integration, the result is:

$$\left[\tau_e A_e \left(\frac{\partial \phi}{\partial x}\right)_e - \tau_w A_w \left(\frac{\partial \phi}{\partial x}\right)_w\right] + \left[\tau_n A_n \left(\frac{\partial \phi}{\partial y}\right)_n - \tau_s A_s \left(\frac{\partial \phi}{\partial y}\right)_s\right] + \bar{S}\Delta V = 0 \qquad \textit{Ec 3.3}$$

A denotes the area of the control volume face. In a two-dimensional equation $A_e = A_w = \Delta y$, $A_n = A_s = \Delta x$. The fluxes across the faces are approximated using the central difference scheme. The source term is linearized as $\bar{S}\Delta V = S_u + S_p \phi_P$. In the end, the equation can be written in this form:

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_S \phi_S + a_N \phi_N + S_u \qquad \textit{Ec 3.4}$$

where:

$$a_E = \frac{\tau_e A_e}{\delta x_E},\ a_W = \frac{\tau_w A_w}{\delta x_W},\ a_N = \frac{\tau_n A_n}{\delta y_N},\ a_S = \frac{\tau_s A_s}{\delta y_S}$$

$$a_P = a_E + a_W + a_S + a_N - S_p$$

Where δx represents the distance of the central node and the neighboring nodes.

At each grid point, the discretized equation is solved and hence the temperature distribution over the domain is obtained. At the boundary nodes, if the first type boundary conditions are given, it can be introduced into the equation by S_u and S_p.

$$S_u = a_B \phi_B;\ S_p = -a_B$$

3.3. Code verification

Code is developed and used to solve two cases.

3.3.1. Case 1

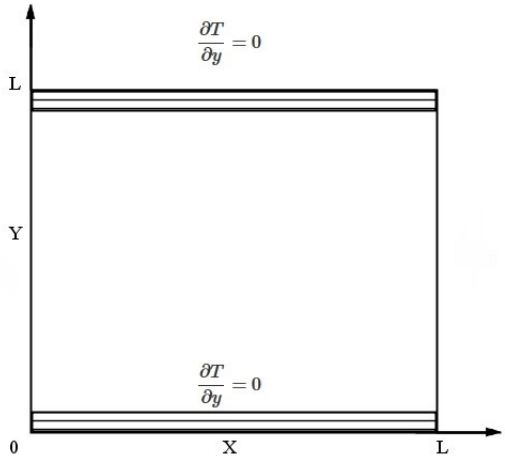


Fig. 3.2. Problem domain for heat conduction

For the first case, a simple heat conduction problem with first type boundary condition is solved. As shown in the picture above, the upper and under wall is adiabatic and the length of the domain L = 1m. For the cold side the temperature is 20°C, as for the hot side, it is 40°C. The heat conduction coefficient is 200 *W/(m²·k)*. The internal heat source is constant, $S_u = 1000\ W/m^3$.

The analytical solution for this case is:

$$\mathrm{T} = -2.5x^2 - 17.5x + 40 \qquad \textit{Ec 3.5}$$

TDMA method is employed to solve this problem. The result of the code with 10, 20, 40 nodes at the middle point of the domain is compared with the analytical solution in the following table:

Table. 3.1 Simulation results of 1-D diffusion problem Case 1

	Analytical	N_{node}		
		10	20	40
T_{middle}	30.625	30.493	30.561	30.593

Here we can see that the results are similar, and with more nodes, the result is more close to the analytical value.

3.3.2. Case 2

For this case, one wall is cooled by a 20 ºC fluid, with heat transfer coefficient of 10 *W/(m²·k)*. The other wall is cooled by a 30 ºC fluid, with heat transfer coefficient of 5 *W/(m²·k)*.

The heat conduction coefficient is also 200 *W/(m^2·k)*, and the length of the domain is 15 m. The internal heat source is determined by the following equation:

$$S_u = 100 \times \frac{x}{L}$$

Where x is the position and L is the length.

The simulation is done with TDMA for N_{node} = 10, 20, 40, 80. The analytical solution is not available in this case. This is only for the purpose of practice the equation with varying source term and convection heat exchange as boundary condition.

The results are shown in the following table:

Table. 3.2 Simulation results of 1-D diffusion problem Case 2

N_{node}	10	20	40	80
T_{middle}	81.983	82.418	82.772	82.976

4. Convection-diffusion equation

Convection is created by fluid flow. It plays an important role in many fluid mechanics and heat transfer problems. In this chapter, the convective term will be included in the presence of a given flow field. The method to obtain the flow field will be introduced in chapter 5.

4.1. Governing equation

Although convection is the only new term introduced in this chapter, since it has an inseparable connection with the diffusion term, the two terms would be handled together. The governing equation is as follow:

$$\frac{\partial(\rho u\phi)}{\partial x}+\frac{\partial(\rho v\phi)}{\partial y}=\frac{\partial}{\partial x}\left(\tau\frac{\partial\phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\tau\frac{\partial\phi}{\partial y}\right)+S_u \qquad \textit{Ec 4.1}$$

4.2. Discretization Schemes

To get the discretized equation, the governing equation is integrated over the control volume:

$$[(\rho u\phi)_w^e]\,dy+[(\rho v\phi)_s^n]\,dx=\left[\left(\tau\frac{\partial\phi}{\partial x}\right)_w^e\right]\mathrm{dy}+\left[\left(\tau\frac{\partial\phi}{\partial y}\right)_s^n\right]\mathrm{dx}+\bar{S}\Delta V \qquad \textit{Ec 4.2}$$

To arrange the equation more compactly, two new symbols *F* and *D* are defined.

$$F_e=(\rho u)_e\Delta y; \qquad F_w=(\rho u)_w\Delta y; \qquad F_n=(\rho u)_n\Delta x; \qquad F_s=(\rho u)_s\Delta x;$$

$$D_e=\frac{\tau_e\Delta y}{(\delta x)_e} \qquad D_w=\frac{\tau_n\Delta y}{(\delta x)_w} \qquad D_n=\frac{\tau_n\Delta x}{(\delta x)_n} \qquad D_s=\frac{\tau_s\Delta x}{(\delta x)_s}$$

Where *F* indicates the strength of the convection, while *D* is the diffusion conductance.

With the substitution of these two symbols, and after using an approximation scheme the discretized equation can then be written as:

$$a_P\phi_P=a_E\phi_E+a_W\phi_W+a_S\phi_S+a_N\phi_N+S_u \qquad \textit{Ec 4.3}$$

It is defined that $[\![A,B]\!]$ denotes the greater of *A* and *B*.

$$a_E=D_eA(|P_e|)+[\![-F_e,0]\!]$$

$$a_W = D_w A(|P_w|) + [\![F_w, 0]\!]$$

$$a_N = D_n A(|P_n|) + [\![-F_n, 0]\!]$$

$$a_S = D_s A(|P_s|) + [\![F_s, 0]\!]$$

The value of $A(|P|)$are different due to different approximation schemes employed.

Table. 4.1 The function $A(|P|)$ for different schemes

Scheme	Formula for $A(\|P\|)$
Central difference	$1 - 0.5P$
Upwind (UPS)	1
Hybrid	$[\![0, 1 - 0.5\|P\|]\!]$
Power law (PLS)	$[\![0, (1 - 0.5\|P\|)^5]\!]$

4.3. Code verification

4.3.1. Case 1: 1-D convection-diffusion

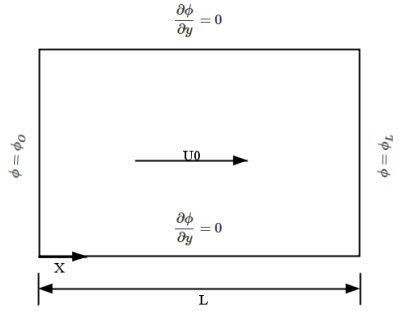


Fig. 4.1. Uniform variation of ϕ in the direction of flow

The problem domain and boundary conditions are as shown in figure above. The horizontal component of velocity has a constant value of *u* over the domain whereas the vertical component is zero. The analytical solution to this problem is given by:

$$\frac{\phi - \phi_0}{\phi - \phi_L} = \frac{exp^{Pex} - 1}{exp^{Pe} - 1} \qquad \textit{Ec 4.4}$$

where Pe is the Peclet number given by:

$$\mathrm{Pe} = \frac{\rho u L}{\tau} \qquad \textit{Ec 4.5}$$

It can be seen that it represents the ratio of the strengths of convection and diffusion.

The results of PLS and UPS and different nodes from the code have been compared with the analytical solution. The maximum error (e_{max}) and average error (e_{avg}) of all nodes are calculated: (where $\phi_0 = 50;\ \phi_L = 100; U = 1; Pe = 1$)

Table. 4.2 Simulation results of 1-D diffusion problem

N_{node}		10	20	40	80	160	320
UPS	e_{max}	0.316	0.155	0.076	0.0379	0.019	0.0094
	e_{avg}	0.208	0.101	0.05	0.025	0.012	0.0062
PLS	e_{max}	0.0013	0.00028	0.00007	0.000016	0.000004	0.000001
	e_{avg}	0.00087	0.00019	0.00004	0.000011	0.000003	0.000001

As we can see in the table, firstly, the error of PLS is generally much smaller than that of UPS. Secondly, every time the number of nodes doubled, the average error and maximum error will decrease by half. While for PLS, it will decrease by quarter. It can be observed much clearer in diagrams below. The slop of the PLS is nearly twice of the UPS. This demonstrates the advantage of PLS over UPS.

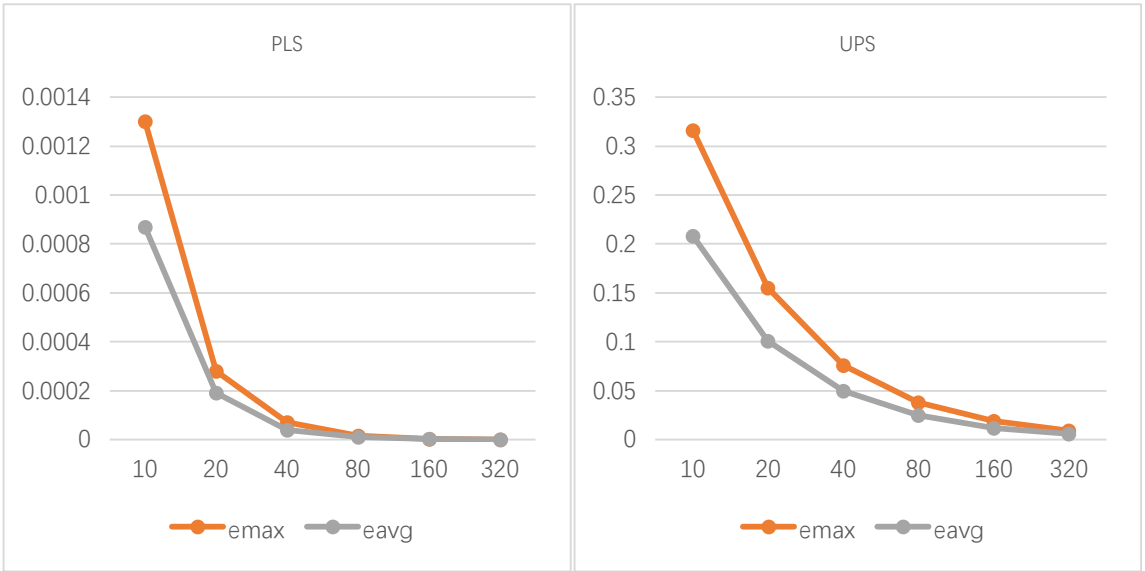


Fig. 4.2. Error Comparison of UPS and PLS

4.3.2. Case 2: Diagonal flux

The problem domain for the diagonal flux case is shown in figure below. This is a case where

the false diffusion effects are clearly observed.

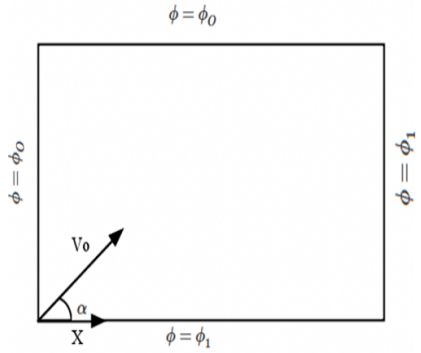


Fig. 4.3. Diagonal flux convection

In this case, the Peclet (*Pe*) number is set very big (as it can be considered infinite), α is set as 45 degrees, and the initial temperature of the domain above the diagonal is set the same, that below the diagonal is set the same as well. Theoretically, the temperature distribution after the simulation should be as follow:

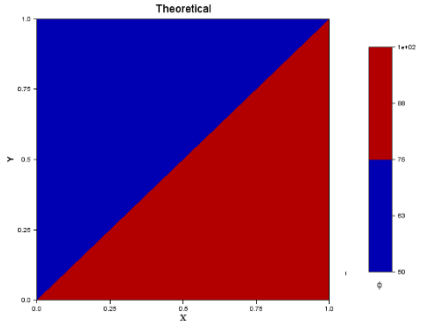


Fig. 4.4. Theoretical Distribution of ϕ

However false diffusion is observed. The temperature distribution does not appear like the theoretical result. With this set of figure below, we can clearly observe the false diffusion and with a refinement of the grid, it can be gradually minimized.

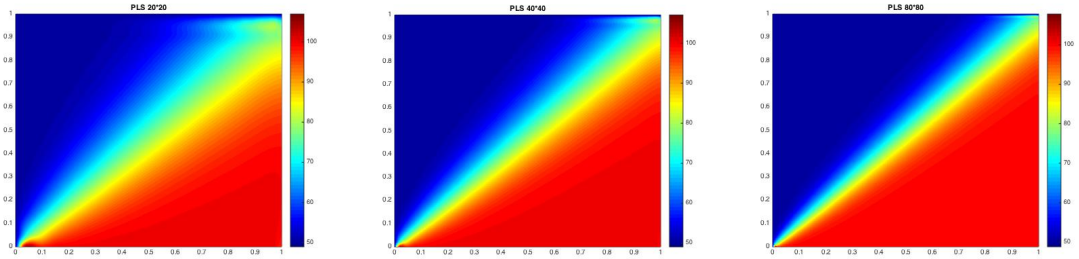


Fig. 4.5. Effect of Grid Size on False Diffusion

A 160*160 grid is built in the domain, and with the employment of PLS, a relatively smaller diffusion is observed. To diminish the false diffusion greatly, a higher resolution scheme need to be employed.

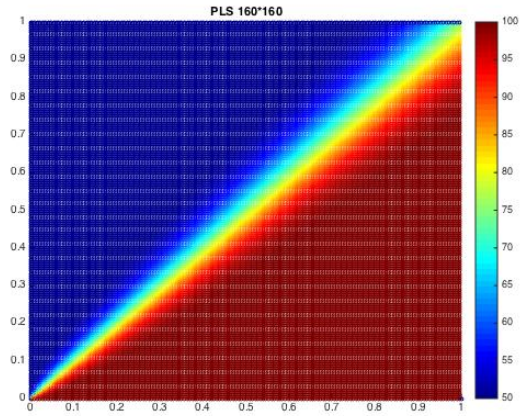


Fig. 4.6. Distribution of ϕ for a 160×160 grid with PLS

4.3.3. Case 3: Solenoidal flux

Another popular test problem for testing various convection models is the solenoidal flux problem [4]. It is also referred to as the Smith- Hutton problem. It consists of a rotating velocity field in the problem domain, as shown in figure below. The

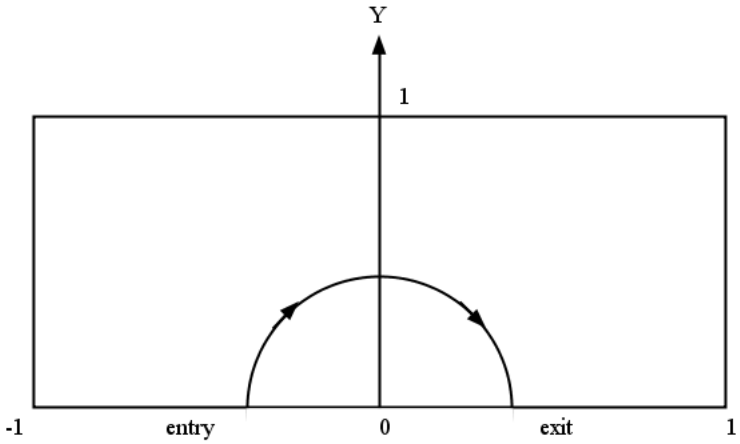


Fig. 4.7. Smith-Hutton Problem (Solenoidal flux problem)

velocity components are given by following equations:

$$\mathrm{u} = 2\mathrm{y}(1-x^2); \qquad \mathrm{v} = -2\mathrm{x}(1-y^2)$$

The boundary conditions are:

$$\phi = 1 - \tanh\beta : \begin{cases} x = -1 & 0 < y < 1 \\ y = 1 & -1 \le x \le 2 \\ x = 1 & 0 < y < 1 \end{cases}$$

$$\phi = 1 + \tanh[\beta(2x+1)] \quad \text{for} \quad y = 0, -1 < x \le 0$$

where β defines the sharpness of the inlet profile. For the case under consideration, $\beta = 10$ radians. At the exit, the second type of boundary consition is imposed:

$$\frac{d\phi}{dy} = 0 \qquad \text{for } y = 0 \quad 0 \le x \le 1$$

Computations were performed for different Peclet number. The results obtained after the grid convergence test are compared with the analytical solution in the following diagrams.

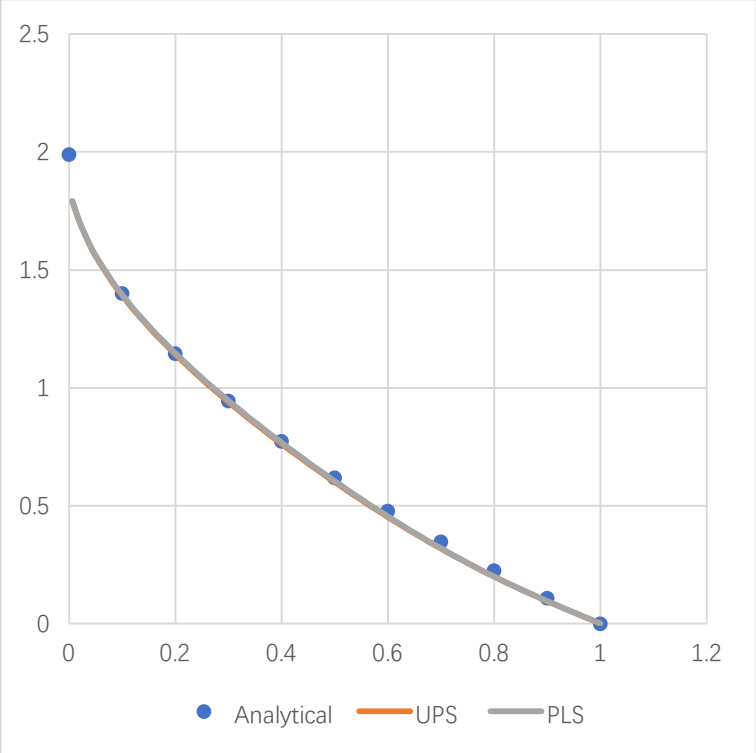


Fig. 4.8. Results for the Smith-Hutton problem at the exit for ρ/λ=10

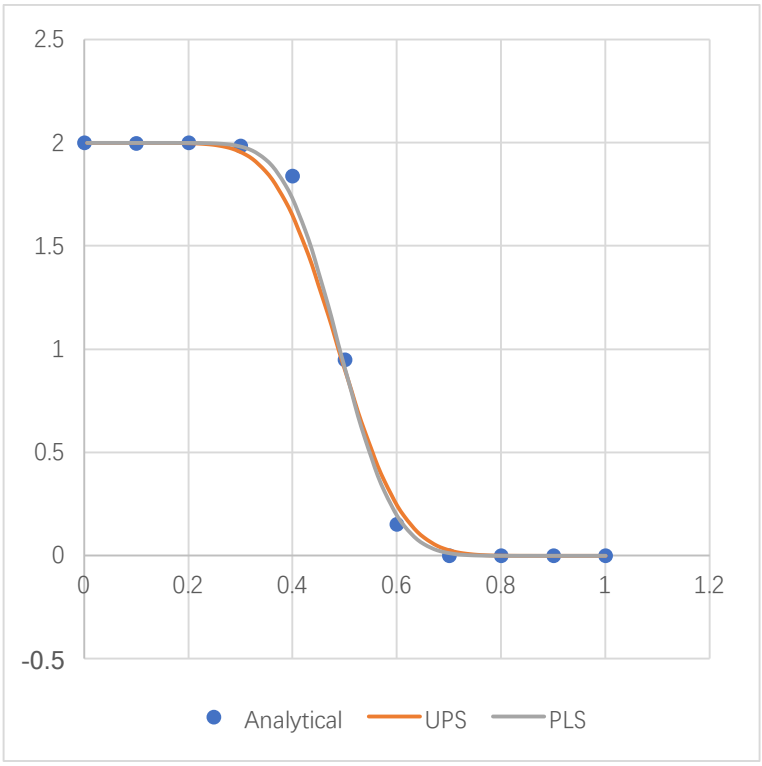


Fig. 4.9. Results for the Smith-Hutton problem at the exit for ρ/λ=1000

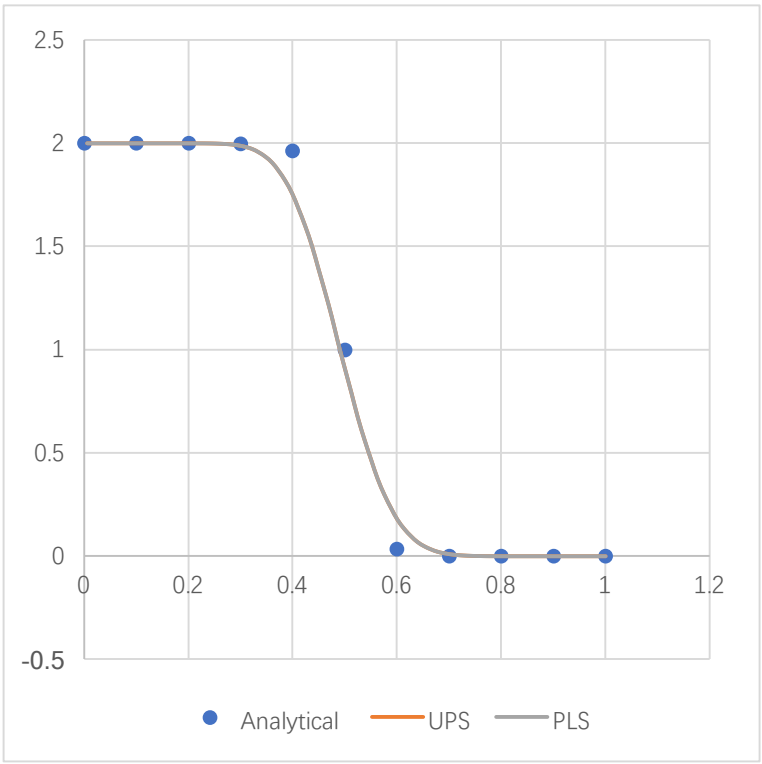


Fig. 4.10. Results for the Smith-Hutton problem at the exit for ρ/λ=1000000

These diagrams demostrates that as the Peclet number increases, the false diffusion errors grow as well. Although PLS is slightly more precise than UPS.

To show the false diffusion, for ρ/λ=1000000, the temperature distribution should be symmetrical, but even with a fine grid (320×160), the false diffusion can still be clearly observed:

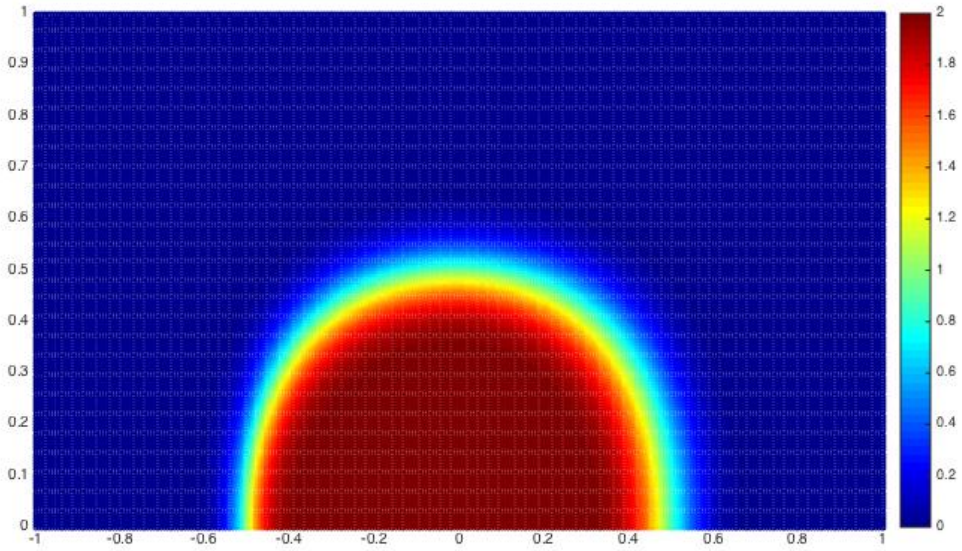


Fig. 4.11. Distribution of ϕ of Smith-Hutton problem at the exit for ρ/λ=1000000

5. Navier-Stokes equation

In the previous chapter, the convection-diffusion equation was solved for a given flow field, while in reality, normally the flow field is unknown. So that it has to be guessed or estimated before it can be used for further computations. The motion of a fluid can be defined by the pressure-velocity coupled continuity and conservation of momentum equations, they are together called Navier-Stokes equations. The flow field can be obtained after solving these equations. In this chapter, the methods for solving the Navier-Stokes equation to obtain a flow field have been given and the code is verified using the benchmark of a lid driven cavity problem.

5.1. Governing equation

Continuity equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0 \qquad \textit{Ec 5.1}$$

X-Momentum equation:

$$\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho u u)}{\partial x} + \frac{\partial (\rho v u)}{\partial y} = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) - \frac{\partial p}{\partial x} + S_u \qquad \textit{Ec 5.2}$$

Y-momentum equation:

$$\frac{\partial (\rho v)}{\partial t} + \frac{\partial (\rho u v)}{\partial x} + \frac{\partial (\rho v v)}{\partial y} = \frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial y}\right) - \frac{\partial p}{\partial x} + S_u \qquad \textit{Ec 5.3}$$

5.2. Solving methods

5.2.1. Staggered grid

When we try to integrate the pressure term in the momentum equation, the discretized momentum equation will contain the pressure difference between two alternate grid points, not between adjacent ones. This will diminish the accuracy of the solution. In 2-D situation, the central pressure of the control volume will have no role to play in the momentum equation. The discretized continuity equation also demands the equality of velocities at alternate grid points and not at adjacent ones. To resolve this difficulty, the model variables may be defined on what are known as staggered grids. It was first used by Harlow and Welch [5]. In this grid, different

variables are defined on different set of grid points.

In the staggered grid, the velocity components are calculated for the points that lie on the faces of the control volumes, as shown in the figure below:

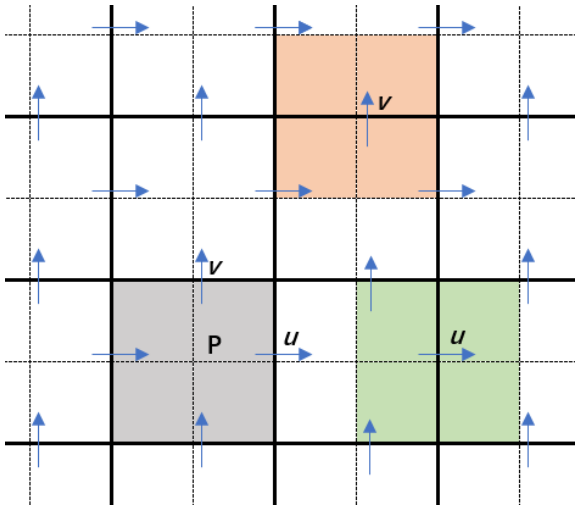


Fig. 5.1. 2-Dimensional staggered grid

With the staggered grid, the mass flow rates across the control-volume faces can be calculated without any interpolation for the relevant velocity component. The discretized continuity equation would contain the differences of adjacent velocity components, and also the pressure difference between two adjacent grid points now becomes the natural driving force for the velocity component located between these grid points.

The momentum equations are discretized on the staggered mesh and are as follows.

Discretized X-Momentum equation:

$$a_{i,j}u_{i,j} = \sum a_{nb}u_{nb} + (p_{i,j} - p_{i+1,j})A_{i,j} + b_{i,j} \qquad \textit{Ec 5.4}$$

Discretized Y-momentum equation:

$$a_{i,j}v_{i,j} = \sum a_{nb}v_{nb} + (p_{i,j} - p_{i,j+1})A_{i,j} + b_{i,j} \qquad \textit{Ec 5.5}$$

where $b_{i,j}$ is the momentum source term.

5.2.2. SIMPLE and SIMPLEC Algorithm

Since the pressure field is unknown, yet we need the pressure field to calculate these velocity components. An iteration process is needed to guess a pressure field first, and then improve the guessed pressure. Some algorithm has been proposed to improve the guessed pressure, and two popular ones are SIMPLE and SIMPLEC.

The order of the execution of the SIMPLE [6] algorithm is as follow:

1. Guess the pressure field *p**.

2. Solve the momentum equation to obtain u*, v*.

$$a_{i,j}u^*_{i,j} = \sum a_{nb}u^*_{nb} + \left(p^*_{i,j} - p^*_{i+1,j}\right)A_{i,j} + b_{i,j}$$

$$a_{i,j}v^*_{i,j} = \sum a_{nb}v^*_{nb} + \left(p^*_{i,j} - p^*_{i,j+1}\right)A_{i,j} + b_{i,j} \qquad \text{Ec 5.6}$$

3. Solve the p´ equation.

$$a_P p'_P = a_E p'_E + a_W p'_W + a_N p'_N + a_S p'_S + b \qquad \text{Ec 5.7}$$

$$\begin{aligned} a_E &= \rho_E d_E \Delta y \\ a_W &= \rho_W d_W \Delta y \\ a_N &= \rho_N d_N \Delta x \\ a_S &= \rho_S d_S \Delta x \end{aligned} \qquad \text{Ec 5.8}$$

$$a_P = a_E + a_W + a_N + a_S \qquad \text{Ec 5.9}$$

$$\mathrm{b} = [(\rho u^*)_e - (\rho u^*)_e]\Delta \mathrm{y} + [(\rho v^*)_e - (\rho v^*)_s]\Delta x \qquad \text{Ec 5.10}$$

Where $d_{i,j} = \frac{A_{i,j}}{a_{i,j}}$. The b term in the pressure correction equation is the mass source calculated using the intermediate velocity fields.

4. Correct pressure by adding p´ to p*.

$$p = p^* + p' \qquad \text{Ec 5.11}$$

5. Calculate u, v using the velocity-correction formulas.

$$u = u^* + \left(p'_{i,j} - p'_{i+1,j}\right)d_{i,j}$$

$$v = v^* + \left(p'_{i,j} - p'_{i,j+1}\right)d_{i,j} \qquad \text{Ec 5.12}$$

6. Solve the discretization equation for the energy equation to obtain the temperature.

7. Treat the corrected pressure as the new guessed pressure, then return to step 2 to repeat the whole procedure.

The pressure-correction equation is easily diverged unless some under-relaxation is imposed. We can relax the velocities or the correction of the pressure by introducing a relaxation factor. If the relaxation factor is not small enough, the result may diverge, but if it is too small, the solution may not progress forward sufficiently enough, and it may oscillate in a small rang. Another way to optimally relax the velocity corrections is by introducing the pseudo-transient approach. In this method, the transient term is introduced, even though a steady state problem is being solved. The transient term acts in a manner similar to that of the relaxation factors being used in the equations. By varying the time step, the equations can be optimally relaxed and hence, a converged solution can be obtained.

SIMPLE was proposed from 1972. It has been super popular in the numerical heat transfer field. Sometimes, for SIMPLE, even with all methods above employed, the results may also diverge. So many researchers have also proposed some improvement for this solver. One of them is employed here in the code developed, which is SIMPLEC [7].

SIMPLEC is based on SIMPLE but improved the correction error in SIMPLE, thus possess a better precision and converge ability.

5.3. Code verification

The 2-D lid driven cavity is one of the most studied and well-known fluid problem in CFD literature. This problem is considered a benchmark for verifying new discretization schemes and solution algorithms.

5.3.1. Geometry

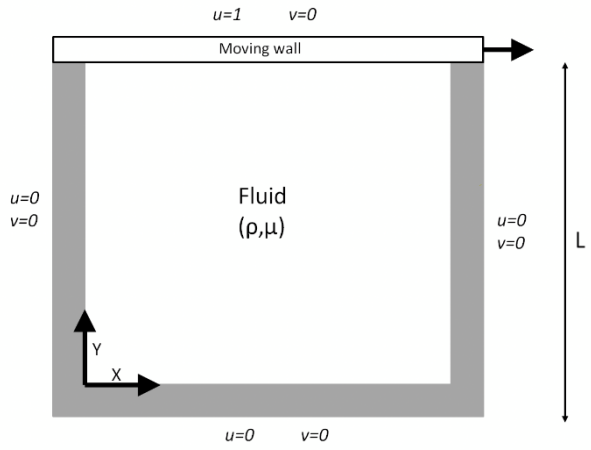


Fig. 5.2. Geometry for the Lid-Driven Cavity Problem

The geometry of the problem is shown in the figure above. The domain is a square, with three static walls. The fluid in this cavity is only driven by the lid moving wall.

5.3.2. Boundary conditions

The length of the cavity is considered as 1 m. The top moving wall only have the x direction velocity constant component. The other three walls are stationary. The Dirichlet boundary conditions are entered into the momentum equations. The no-slip condition is applied at all the walls. The boundary conditions are illustrated as follows:

$$u(x,0) = u(0,y) = u(L,y) = 0$$

$$u(x,L) = 1$$

$$v(x,0) = v(x,L) = v(0,y) = v(L,y) = 0$$

At x=0 and x=L:

$$\frac{\partial p}{\partial x} = 0$$

At y=0 and y=L:

$$\frac{\partial p}{\partial y} = 0$$

5.4. Solving methods

5.4.1. With developed code

For the discretization of the convection term, the UPS and PLS have been used. The diffusion terms are discretized by the central difference method. Any properties not available at the face or node being considered are calculated using linear interpolation.

The pseudo-transient approach is used for relaxing the velocity terms. For every case considered, a suitable time-step is chosen based on the Reynolds number and mesh spacing to ensure stability and that the solution converges. For solving the system of equations, one iteration of SOR is used in the X and Y direction for the discretized momentum equations. For solving the pressure correction equation, since it will be corrected later, the residual is only reduced to less than 0.1. The residuals both velocity components are reduced to 10^{-8}.

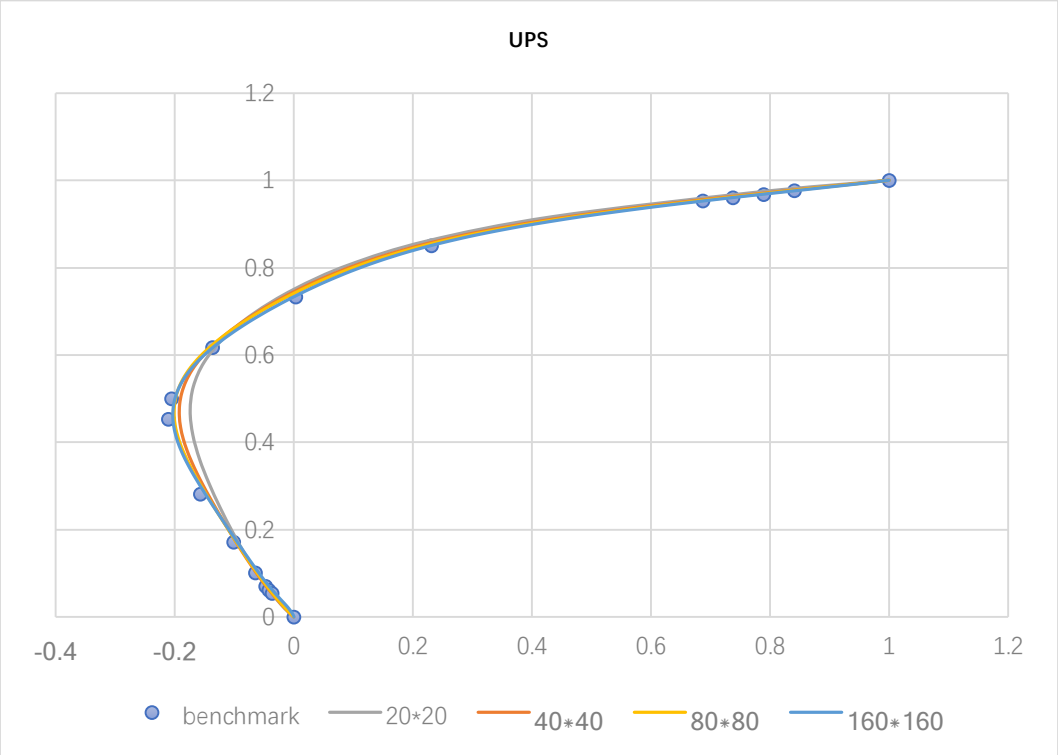
5.4.2. With ANSYS Fluent

In the ANSYS simulation, 18818 nodes, 9216 elements are used. All parameters are adjusted strategically to satisfy the Reynolds number requirement. The SIMPLE method is used here, and second order upwind scheme is used to solve the momentum equation. Other parameters are set as the same of the code.

5.5. Results of the code developed

The problem is solved for different values of the Reynolds number (Re). For each Re, grid convergence study is performed and the results have been shown in the velocity plots. The solutions obtained from the code have been compared to the benchmark solution of Ghia et al [8], which is the most widely used benchmarks for this case.

5.5.1. Re = 100



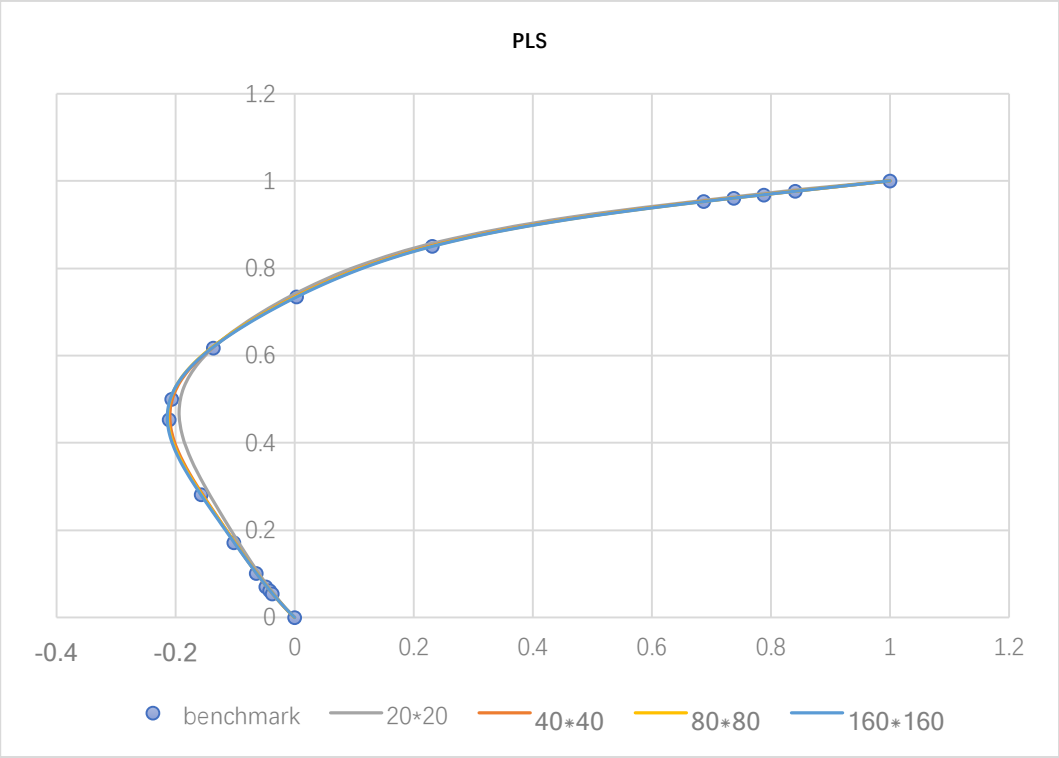
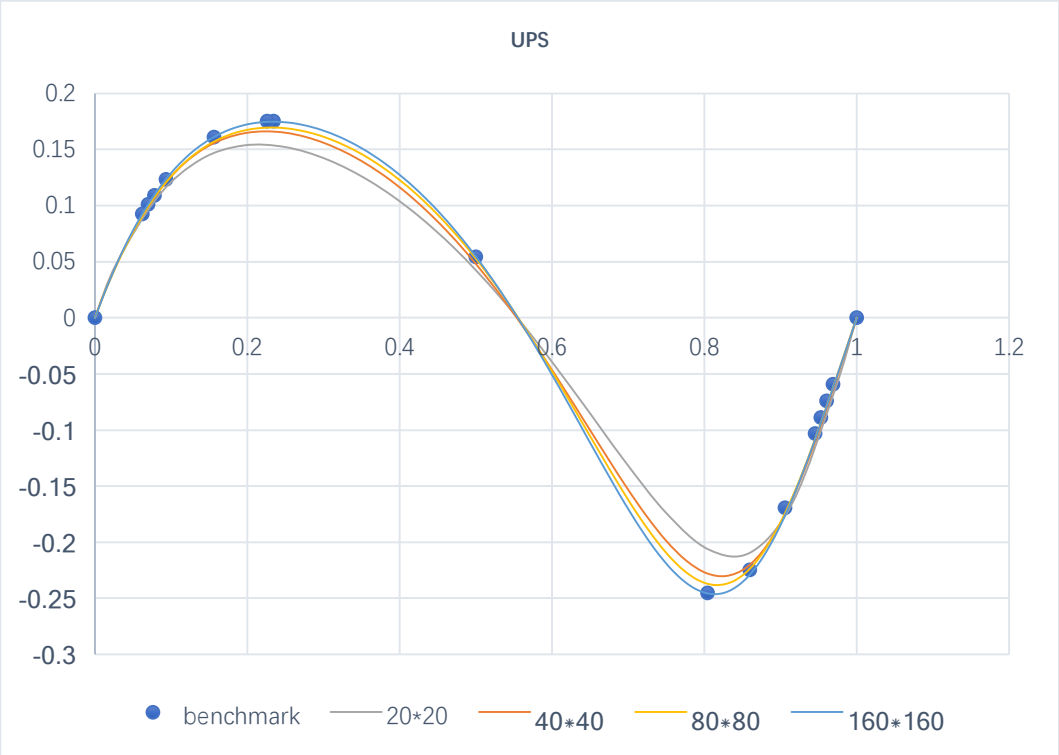


Fig. 5.3. Results for *u* along Vertical Line through Geometric Centre for Re = 100



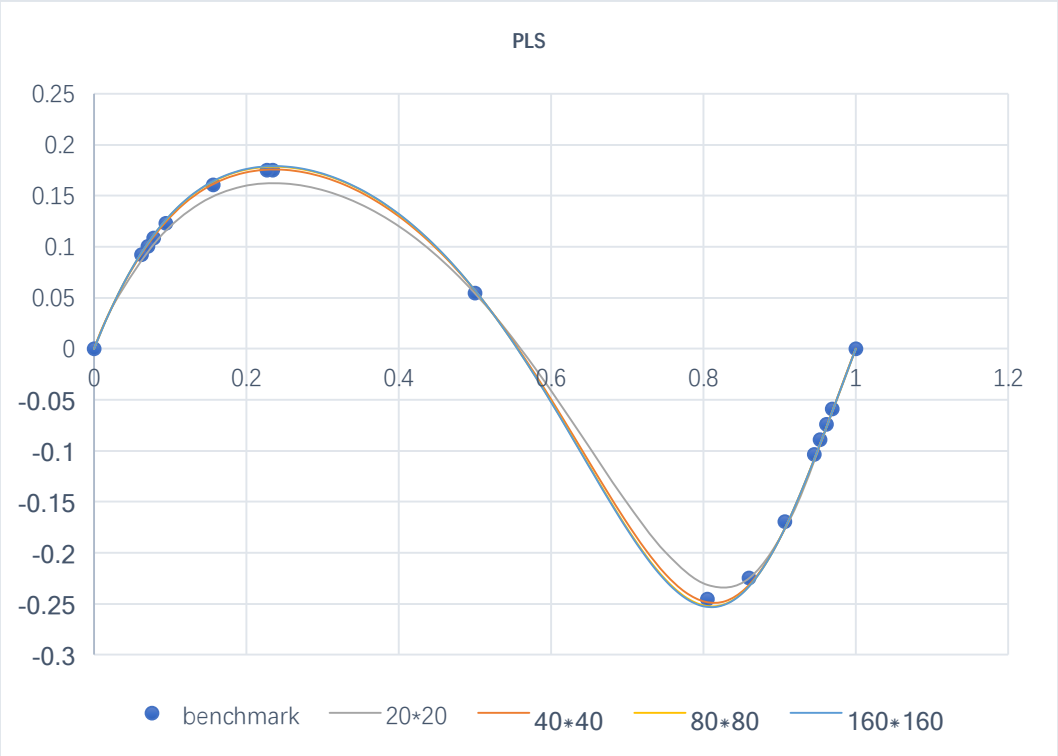


Fig. 5.4. Results for *v* along Horizontal Line through Geometric Centre for Re = 100

The results above show a little bit difference from the benchmark. Since different shcemes and solvers are employed in this simulation, the difference of the results is considered very small and acceptable.

As seen from the above figures, the limitations of the upwind scheme is clearly visible as the power law scheme gives a good solution in agreement with the benchmark on a coarser grid of 40×40, whereas for the upwind scheme, it requires a so much finer mesh like 160×160.

5.5.2. Re = 400

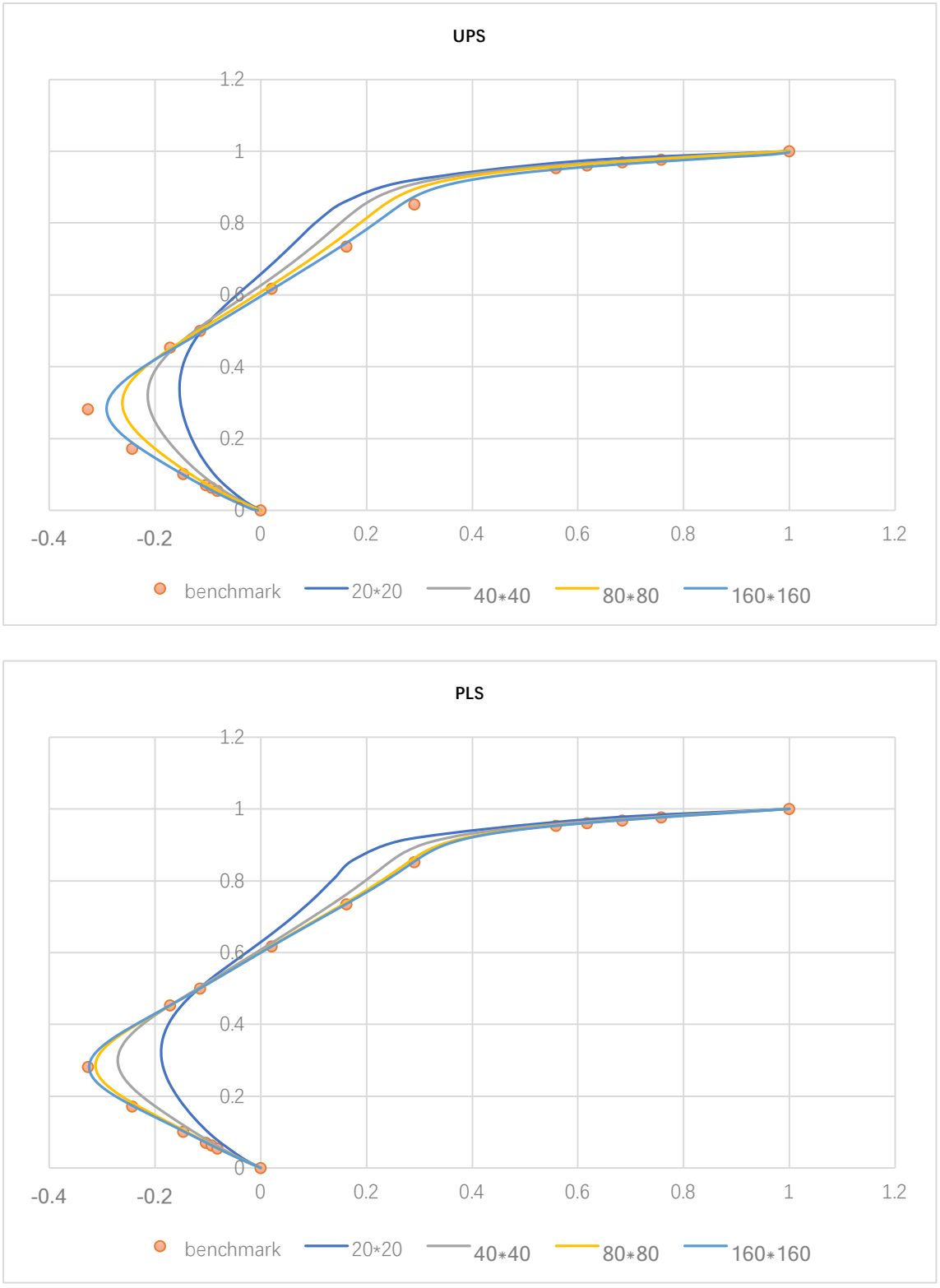
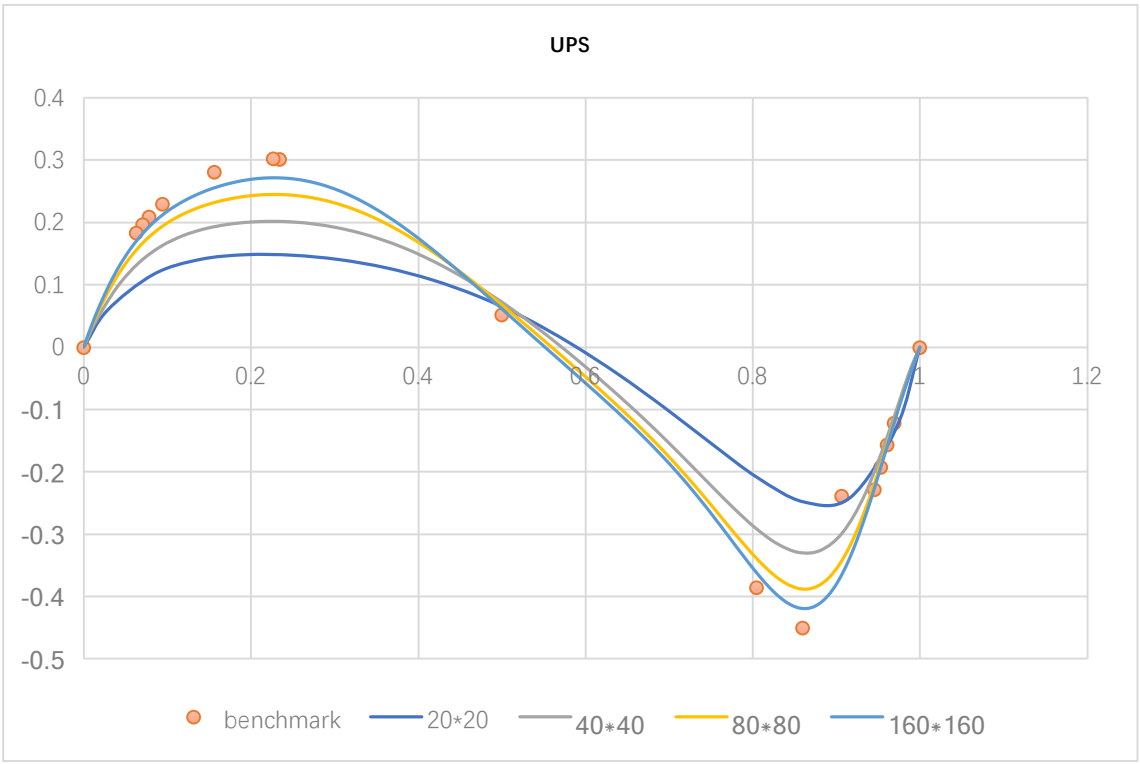


Fig. 5.5. Results for *u* along Vertical Line through Geometric Centre for Re = 400



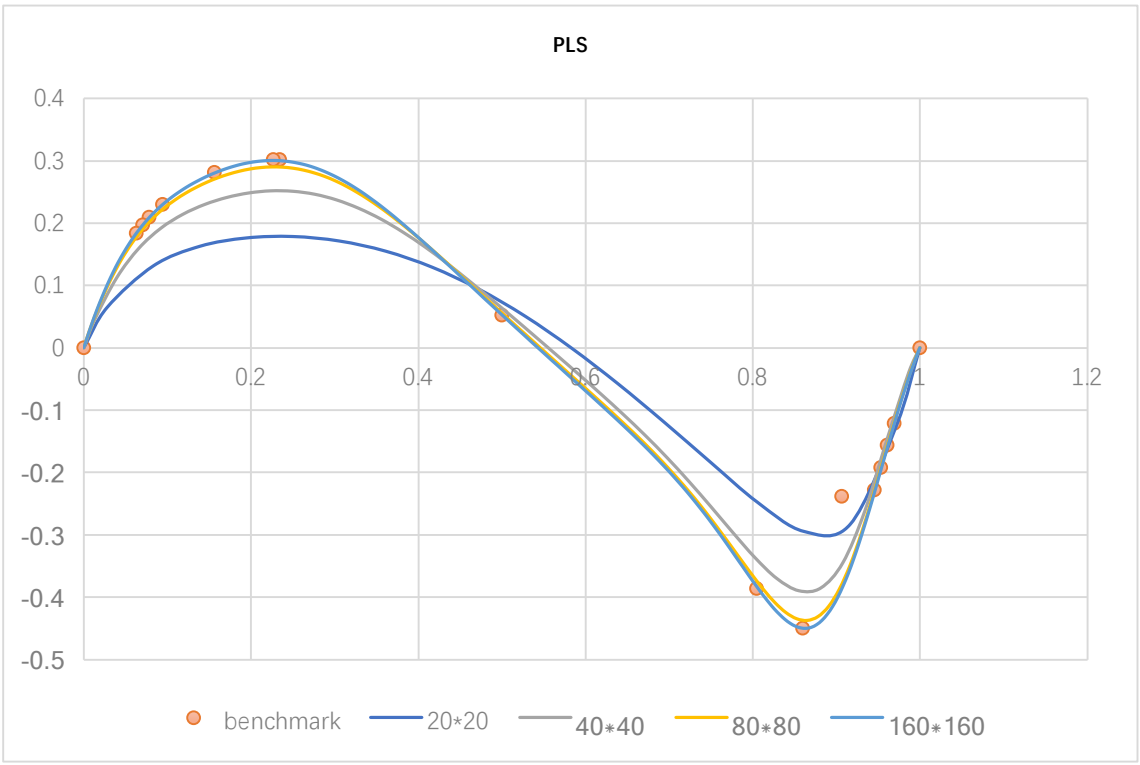


Fig. 5.6. Results for *v* along Horizontal Line through Geometric Centre for Re = 400

5.5.3. Re = 1000

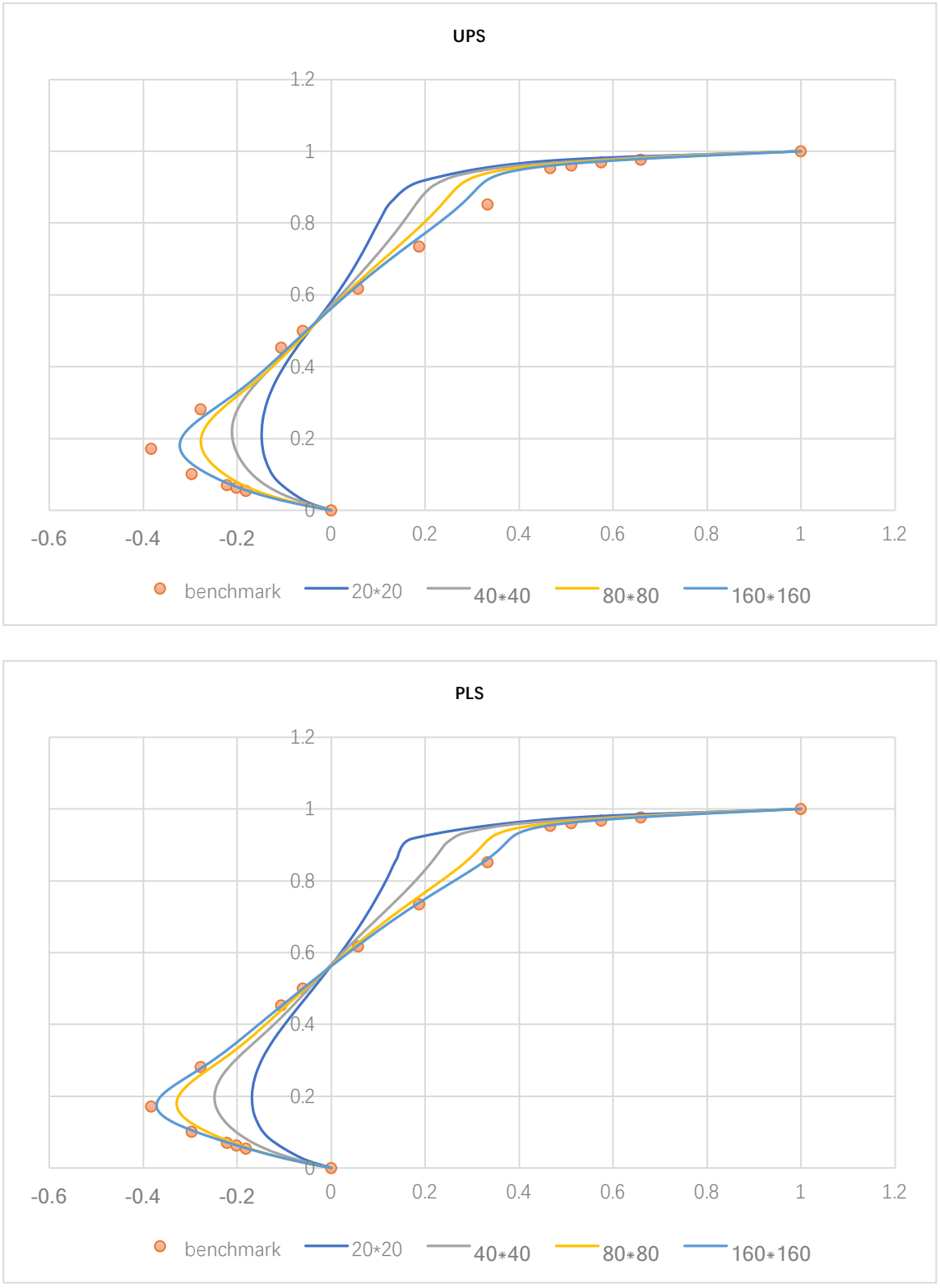


Fig. 5.7. Results for *u* along Vertical Line through Geometric Centre for Re = 1000

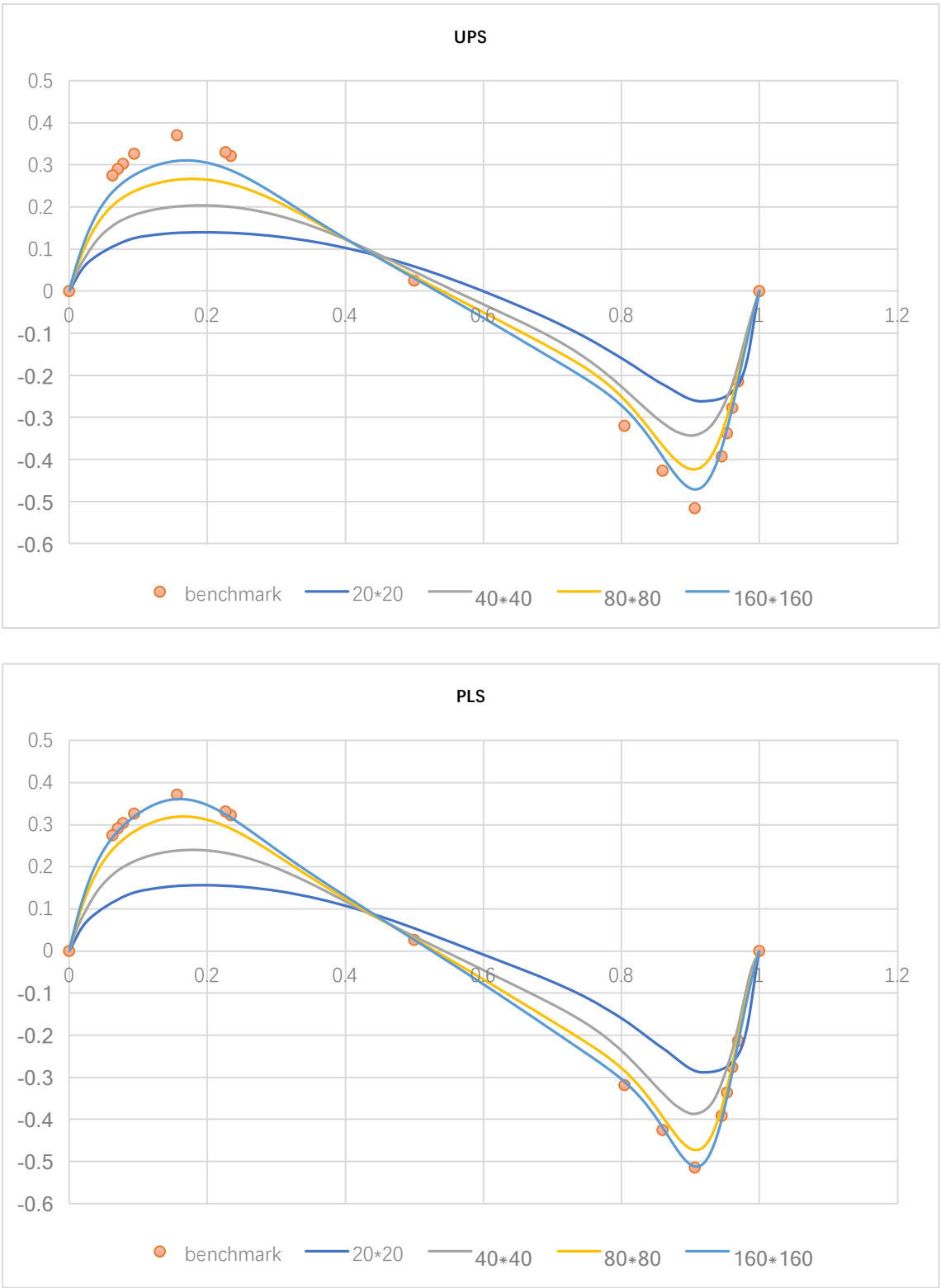


Fig. 5.8. Results for *v* along Horizontal Line through Geometric Centre for Re = 1000

With the increasing Reynolds number, it takes more computational time and relax method for converging the solution. It can be observed that, with bigger Reynolds number, the results of the simulation are marching to the benchmark so much slower from the coarse mesh to fine mesh, and the advantage of PLS is more obvious.

Overall, the code developed can generate great results with little difference from the benchmark. Although, some parameters would need to be adjusted to get converged results.

5.6. Results of ANSYS Fluent

Results of ANSYS are plotted together with the results of the code (160×160 with PLS) and the benchmark for different Reynolds number.

5.6.1. Re = 100

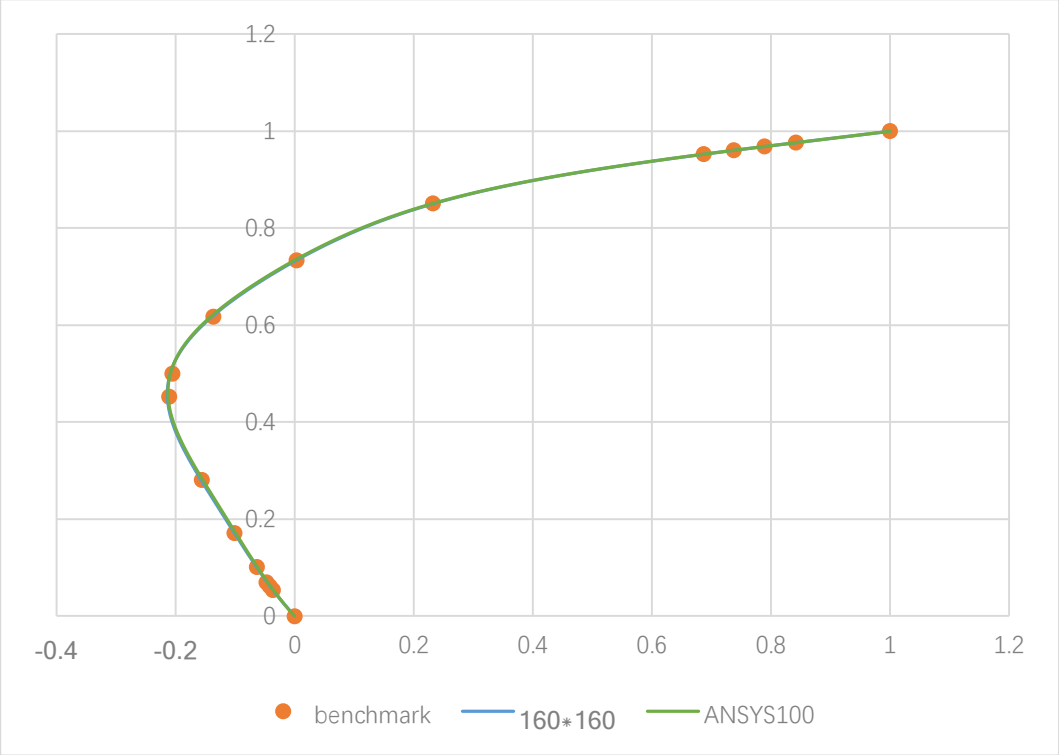


Fig. 5.9. Comparison for *u* along Vertical Line through Geometric Centre for Re = 100 with ANSYS

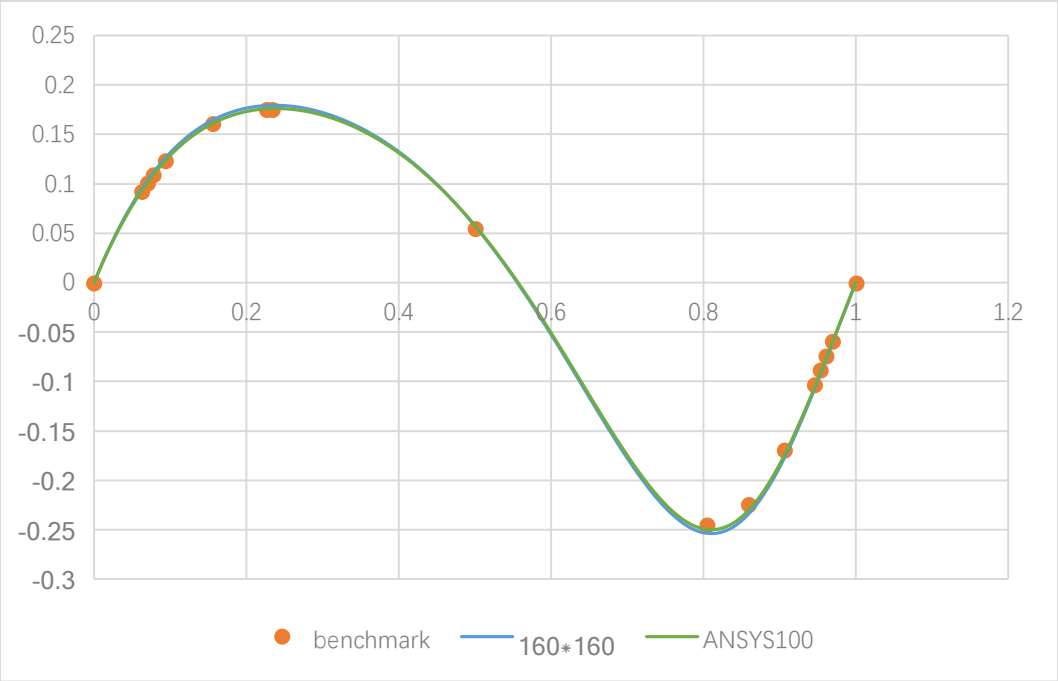


Fig. 5.10. Comparison for *v* along Horizontal Line through Geometric Centre for Re = 100 with ANSYS

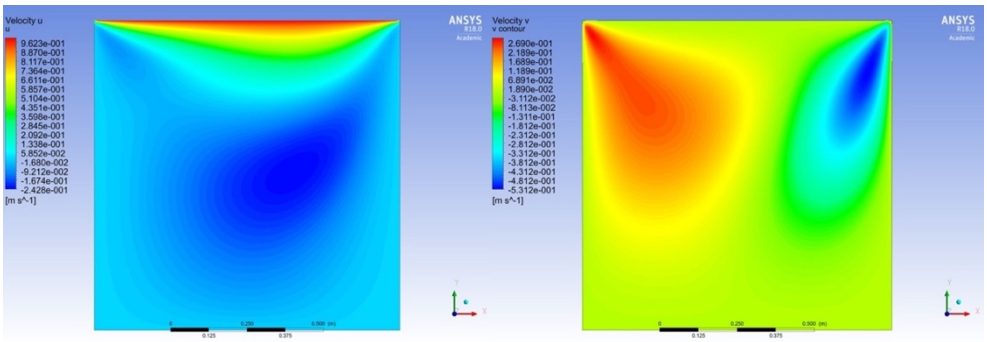


Fig. 5.11. Distribution of *u* and *v* over the cavity for Re = 100

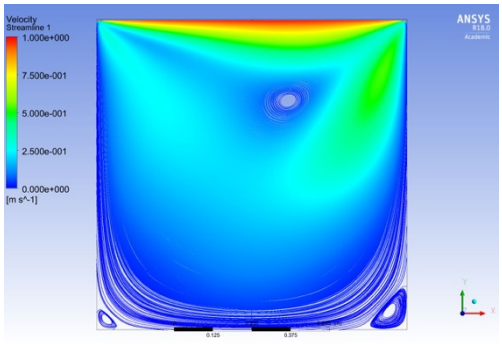


Fig. 5.12. Streamline Contours for Re = 100

5.6.2. Re = 400

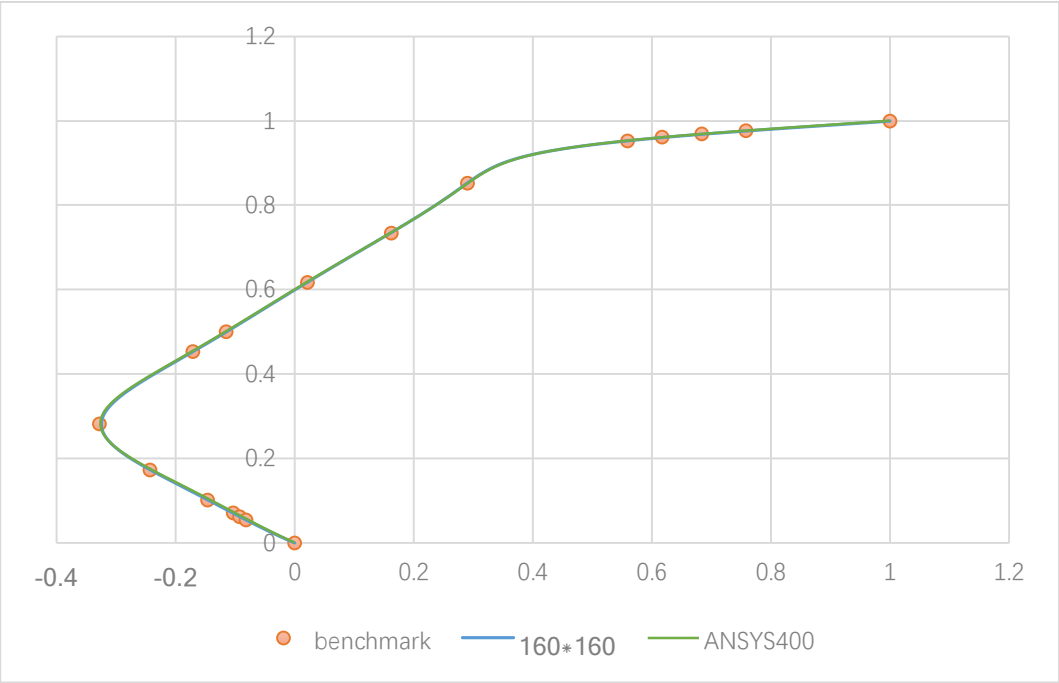


Fig. 5.13. Comparison for *u* along Vertical Line through Geometric Centre for Re = 400 with ANSYS

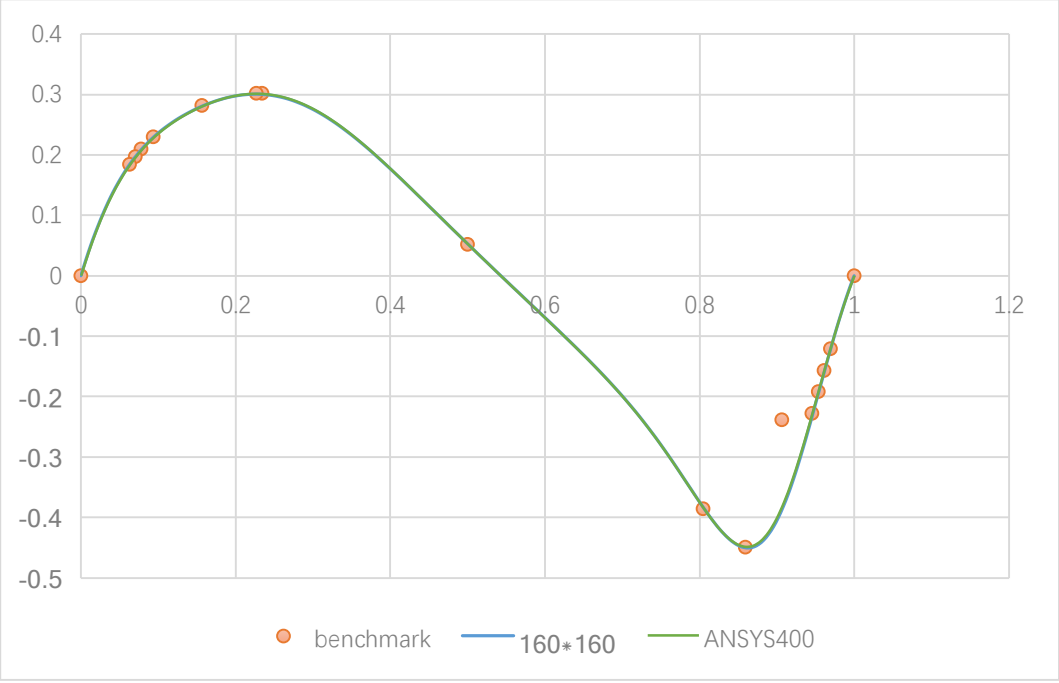


Fig. 5.14. Comparison for *v* along Horizontal Line through Geometric Centre for Re = 400 with ANSYS

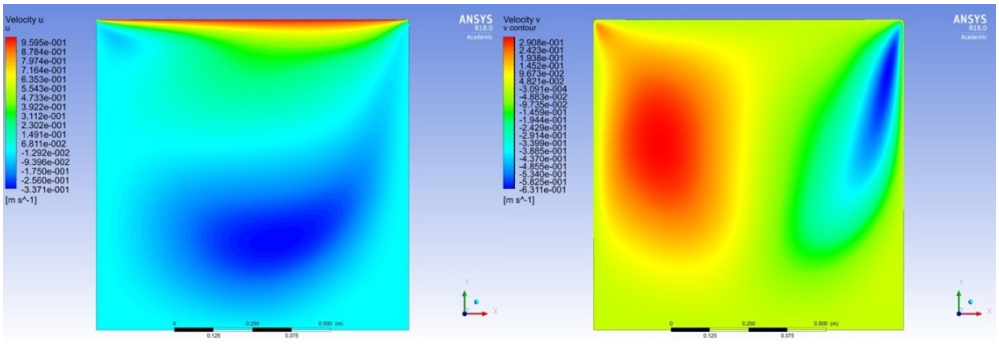


Fig. 5.15. Distribution of *u* and *v* over the cavity for Re = 400

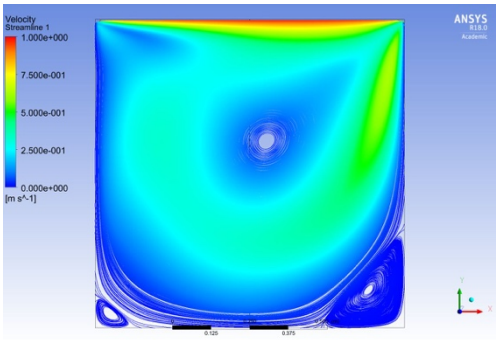


Fig. 5.16. Streamline Contours for Re = 400

5.6.3. Re = 1000

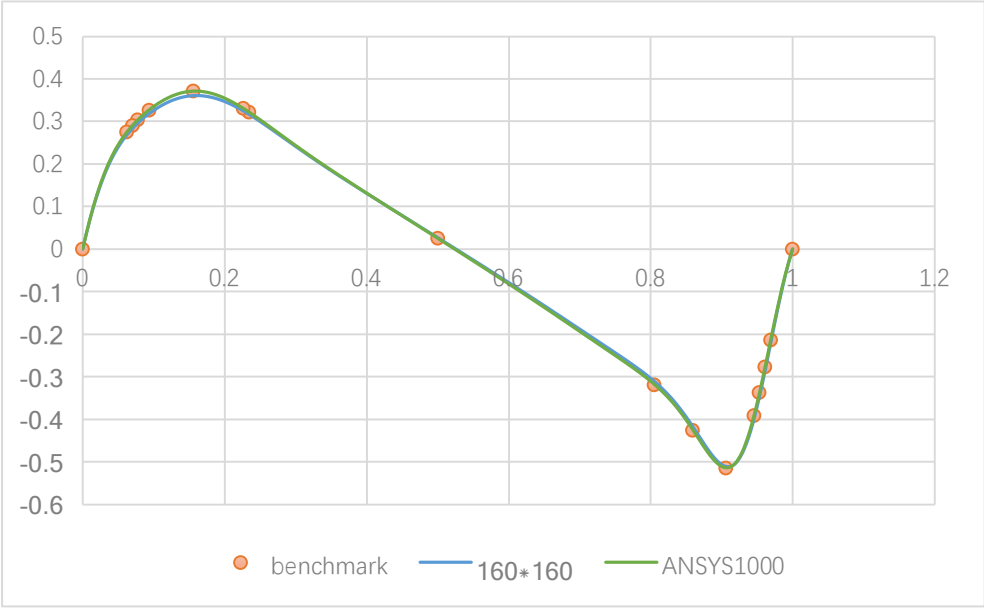


Fig. 5.17. Comparison for *u* along Vertical Line through Geometric Centre for Re = 1000 with ANSYS

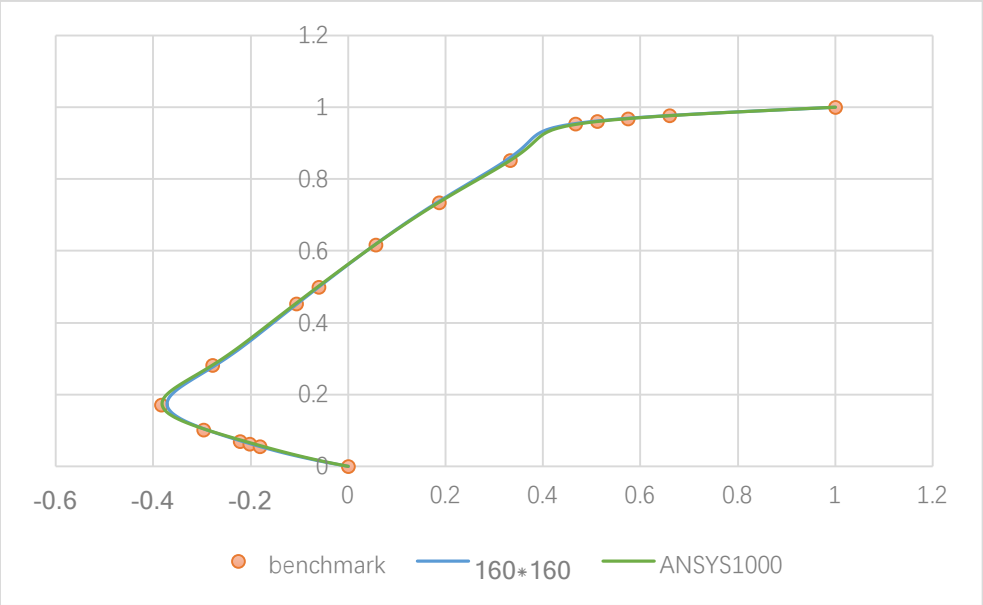


Fig. 5.18. Comparison for *v* along Horizontal Line through Geometric Centre for Re = 1000 with ANSYS

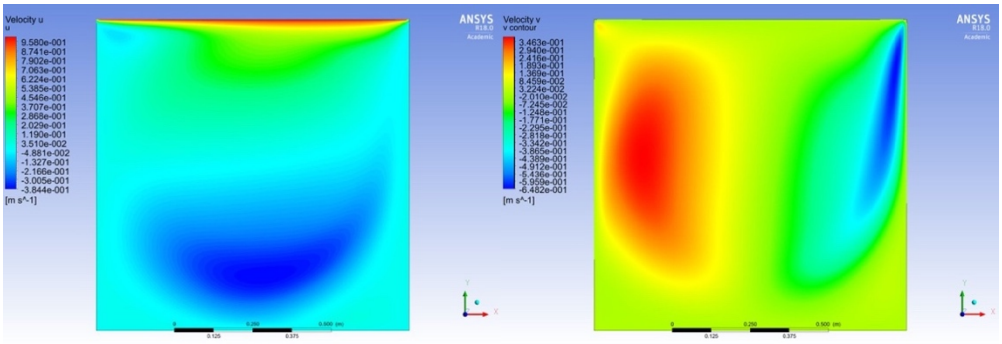


Fig. 5.19. Distribution of *u* and *v* over the cavity for Re = 1000

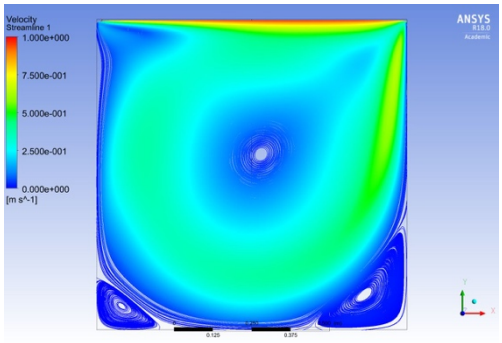


Fig. 5.20. Streamline Contours for Re = 1000

5.6.4. Re = 3200

For ANSYS, a computation for an even higher Reynolds number was done.

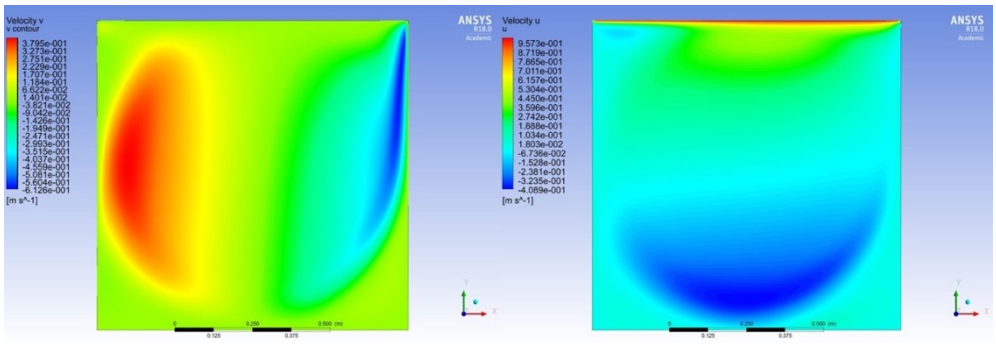


Fig. 5.21. Distribution of *u* and *v* over the cavity for Re = 3200

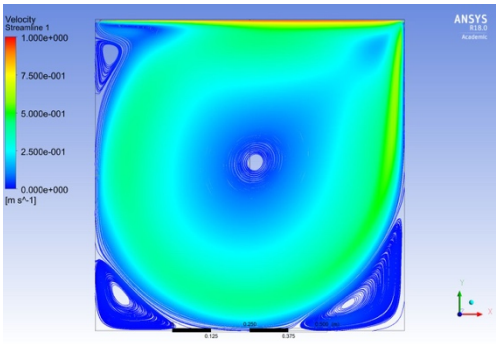


Fig. 5.22. Streamline Contours for Re = 3200

From the plotted diagrams of velocities, we can see that there are little differences between the results of code developed and that of ANSYS.

The Reynolds number is the ratio of inertial forces to viscous forces. From the velocity distribution figures and streamline contours, it can be observed that, as the Reynolds number grows, the velocity change of the fluid is easier to develop. Thus, for higher Re, we can see larger number of velocity, and two bigger vortexes in the streamline contours pictures are formed in the bottom sides of the cavity, and the higher the Re, the bigger the vortexes.

6. Navier-Stokes equation and energy equation

6.1. Governing equation

An energy conservation equation is added in this case.

Continuity equation:

$$\frac{\partial \rho}{\partial \bar{t}} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \qquad \textit{Ec 6.1}$$

X-Momentum equation:

$$\frac{\partial(\rho u)}{\partial \bar{t}} + \frac{\partial(\rho u u)}{\partial x} + \frac{\partial(\rho v u)}{\partial y} = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) - \frac{\partial p}{\partial x} + S_u \qquad \textit{Ec 6.2}$$

Y-momentum equation:

$$\frac{\partial(\rho v)}{\partial \bar{t}} + \frac{\partial(\rho u v)}{\partial x} + \frac{\partial(\rho v v)}{\partial y} = \frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial y}\right) - \frac{\partial p}{\partial y} + S_u + B_y \qquad \textit{Ec 6.3}$$

Energy equation:

$$\frac{\partial(\rho c_p T)}{\partial \bar{t}} + \frac{\partial(\rho c_p u T)}{\partial x} + \frac{\partial(\rho c_p v T)}{\partial y} = \frac{\partial}{\partial x}\left(\kappa \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\kappa \frac{\partial T}{\partial y}\right) + S_u \qquad \textit{Ec 6.4}$$

where, B_y is the body force to account for the buoyancy in the problem.

6.1.1. The Boussinesq approximation

When the temperatures change across the cavity, then the temperature dependent properties such as the density, viscosity also change. Since density plays a significant role in the buoyancy term at each node, the variation of density due to the change of temperature cannot be ignored. While computationally, it is very complicated to account for the change in density everywhere, as the density term appears at many places in the momentum equation. As a result, in order to simplify the problem while accounting for the density changes, the Boussinesq approximation is used.

It was first proposed by Joseph Valentin Boussinesq. The detail of this approximation can be found in [9]. Under this approximation, the density variations can be neglected everywhere in

the momentum term, except the buoyancy term. In this way, the Navier-Stokes equation is greatly simplified and computations become easier, while taking into account the density changes as well. The density in buoyancy term is then approximated as:

$$\rho = \rho_0(1 - \beta\Delta T) \qquad \textit{Ec 6.5}$$

Where β is the e coefficient of volumetric thermal expansion.

6.1.2. Non-dimensionalization

The three dimensionless numbers that characterize the problem are:

Prandtl Number

The Prandtl number (Pr) is the ratio of the momentum diffusivity to the thermal diffusivity.

$$Pr = \frac{\upsilon}{\alpha} \qquad \textit{Ec 6.6}$$

Where υ is the kinematic viscosity, and α is the thermal diffusivity.

Grashof number

The Grashof number (Gr) is a dimensionless number in fluid dynamics and heat transfer which approximates the ratio of the buoyancy to viscous force acting on a fluid.

$$Gr = \frac{g\beta\Delta T l^3}{\upsilon^2} \qquad \textit{Ec 6.7}$$

- g is gravitational acceleration
- β is the coefficient of thermal expansion
- ΔT is the temperature difference
- l is the geometry length

Rayleigh Number

Rayleigh number (Ra) is the product of the Grashof number and Prandtl numbers. When the Rayleigh number is below a critical value for that fluid, heat transfer is primarily in the form of conduction; when it exceeds the critical value, heat transfer is primarily in the form of convection.

$$Ra = Gr \cdot \mathrm{Pr} = \frac{g\beta\Delta T l^3}{\upsilon\alpha}$$

In order to control the simulation only by Ra and Pr, the governing equations given are non-dimensionalized using the following substitutions:

$$X = \frac{x}{L}; \qquad Y = \frac{y}{L}; \qquad U = \frac{uL}{\alpha}; \qquad V = \frac{vL}{\alpha}; \qquad t = \frac{\tilde{t}\alpha}{L^2};$$

$$P = \frac{(p-\rho g L)L^2}{\rho\alpha^2}; \qquad \theta = \frac{T-T_c}{T_H-T_c}$$

After using the above substitutions, the governing equations in the non-dimensional form are:

Continuity equation:

$$\frac{\partial}{\partial t} + \frac{\partial(U)}{\partial x} + \frac{\partial(V)}{\partial y} = 0 \qquad \textit{Ec 6.8}$$

X-Momentum equation:

$$\frac{\partial(U)}{\partial t} + \frac{\partial(UU)}{\partial x} + \frac{\partial(VU)}{\partial y} = \frac{\partial}{\partial x}\left(Pr\frac{\partial U}{\partial x}\right) + \frac{\partial}{\partial y}\left(Pr\frac{\partial U}{\partial y}\right) - \frac{\partial P}{\partial x} + S_u \qquad \textit{Ec 6.9}$$

Y-momentum equation:

$$\frac{\partial(V)}{\partial t} + \frac{\partial(UV)}{\partial x} + \frac{\partial(VV)}{\partial y} = \frac{\partial}{\partial x}\left(Pr\frac{\partial V}{\partial x}\right) + \frac{\partial}{\partial y}\left(Pr\frac{\partial V}{\partial y}\right) - \frac{\partial P}{\partial x} + S_u + RaPr\theta \qquad \textit{Ec 6.10}$$

Energy equation:

$$\frac{\partial(\theta)}{\partial t} + \frac{\partial(U\theta)}{\partial x} + \frac{\partial(V\theta)}{\partial y} = \frac{\partial}{\partial x}\left(\frac{\partial\theta}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\partial\theta}{\partial y}\right) + S_u \qquad \textit{Ec 6.11}$$

6.2. Problem description

6.2.1. Geometry

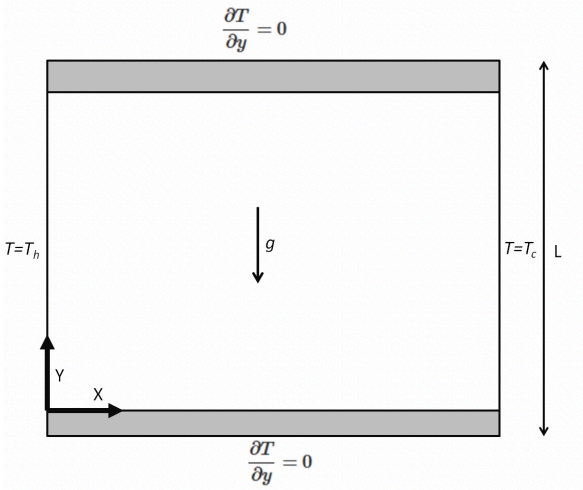


Fig. 6.1. Geometry for the Differentially Heated Cavity Problem

The geometry for the problem has been shown in the above figure. The square cavity is of length L. The two horizontal walls are adiabatic. The left vertical walls are the hot wall; the right vertical wall is the cold wall. They are both maintained a constant temperature.

6.2.2. Boundary conditions

The boundary conditions for the problem are as follows:

$$u(x,0) = u(0,y) = u(L,y) = u(x,L) = 0$$

$$v(x,0) = v(x,L) = v(0,y) = v(L,y) = 0$$

$$At\ x = 0: \quad T(0,y) = T_h \Rightarrow \theta(0,y) = 1$$

$$At\ x = L: \quad T(L,y) = T_c \Rightarrow \theta(L,y) = 0$$

At y = 0 and y = L:

$$\frac{\partial T}{\partial y} = 0 \Rightarrow \frac{\partial \theta}{\partial y} = 0$$

At x = 0 and x = L:

$$\frac{\partial p}{\partial x} = 0$$

At y = 0 and y = L:

$$\frac{\partial p}{\partial y} = 0$$

6.3. Solving methods

6.3.1. With developed code

This problem is solved using pretty much the same method as the lid driven cavity problem, except that in this case, the solver of the pressure correction equation is iterated 10 times in every velocity field iteration. After every correction of the velocity field, the energy equation is solved once to get the temperature distribution.

The Pr is set as 0.71 in order to compare with the benchmark. For each different Ra (10^3, 10^4, 10^5), meshes of 40×40, 80×80, 160×160 are used to run the simulation.

6.3.2. With ANSYS Fluent

In ANSYS, energy model was turned on to do the heat transfer simulation. SIMPLE method is used here, and for all pressure, momentum and energy discretization, second order scheme are employed. Different parameters (including the gravitational acceleration, operation conditions and fluid properties) are introduced strategically to set Pr as 0.71 and Ra as 10^3, 10^4, 10^5 and 10^6. For the density of the fluid, the Boussinesq approximation is used.

6.4. Results of simulations

6.4.1. Benchmark comparison

The solutions from the code and ANSYS are compared with the benchmark of Vahl Davis [10] for the following quantities:

• u_{max}: The maximum value of horizontal velocity along the vertical line through geometric center of cavity

• y_u: Position of u_{max} along the vertical line through the geometric center of cavity

• v_{max}: The maximum value of vertical velocity along the horizontal line through geometric center of cavity

• x_v: Position of v_{max} along the horizontal line through the geometric center of cavity

• Nu_{max}: The maximum value of local Nu along the hot wall at x = 0

• y_{Numax}: Position of Nu_{max} along the hot wall at x = 0

• y_{Numin}: Position of Nu_{min} along the hot wall at x = 0

• Nu_{min}: The minimum value of local Nu along the hot wall at x = 0

• Nu_{avr}: The value of average Nu along the hot wall at x = 0

Table. 6.1 Results for Ra= 1000 using ANSYS and Developed Code

	Benchmark	ANSYS	Developed code		
			160×160	80×80	40×40
x_v	0.178	0.176	0.180	0.173	0.171
v_{max}	3.697	3.678	3.681	3.664	3.636
y_u	0.813	0.814	0.813	0.814	0.803
u_{max}	3.649	3.635	3.633	3.618	3.582
y_{Nu}	0.092	0.094	0.092	0.083	0.092
Nu_{max}	1.505	1.496	1.510	1.513	1.523
y_{Numin}	1.0	1.0	1.0	1.0	1.0
Nu_{min}	0.692	0.678	0.696	0.700	0.710
Nu_{avr}	1.118	1.112	1.121	1.124	1.130

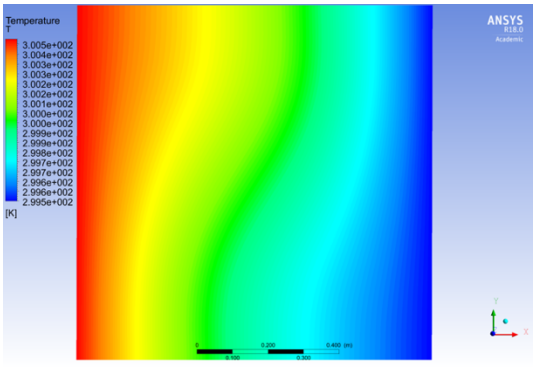
Table. 6.2 Results for Ra= 10000 using ANSYS and Developed Code

	Benchmark	ANSYS	Developed code		
			160×160	80×80	40×40
x_v	0.119	0.121	0.117	0.122	0.118
v_{max}	19.620	19.583	19.663	19.605	19.622
y_u	0.823	0.824	0.826	0.827	0.829
u_{max}	16.180	16.221	16.197	16.161	16.154
y_{Nu}	0.130	0.135	0.142	0.135	0.145
Nu_{max}	3.528	3.512	3.545	3.557	3.607
y_{Numin}	1.0	1.0	0.589	1.0	1.0
Nu_{min}	0.586	0.576	2.250	0.589	0.594
Nu_{avr}	2.243	2.240	0.117	2.248	2.257

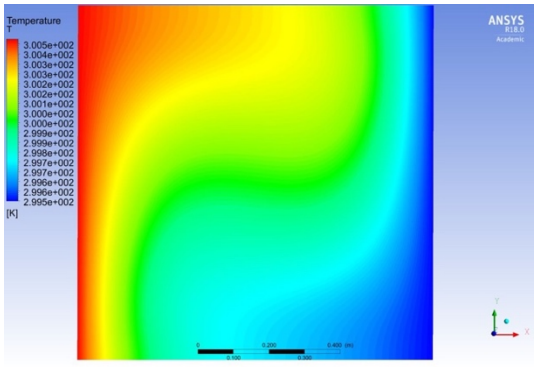
Table. 6.3 Results for Ra= 100000 using ANSYS and Developed Code

	Benchmark	ANSYS	Developed code		
			160×160	80×80	40×40
x_v	0.066	0.065	0.067	0.071	0.066
v_{max}	68.590	68.145	70.613	68.398	68.937
y_u	0.855	0.854	0.858	0.853	0.855
u_{max}	34.730	34.981	35.796	34.833	35.271
y_{Nu}	0.081	0.083	0.079	0.071	0.066
Nu_{max}	7.177	7.754	7.936	7.895	8.288
y_{Numin}	1.0	1.0	1.0	1.0	1.0
Nu_{min}	0,729	0.730	0.786	0.740	0.759
Nu_{avr}	4.519	4.529	4.681	4.547	4.615

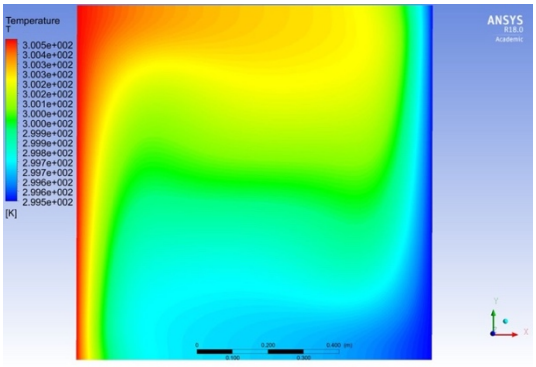
6.4.2. Thermal distribution



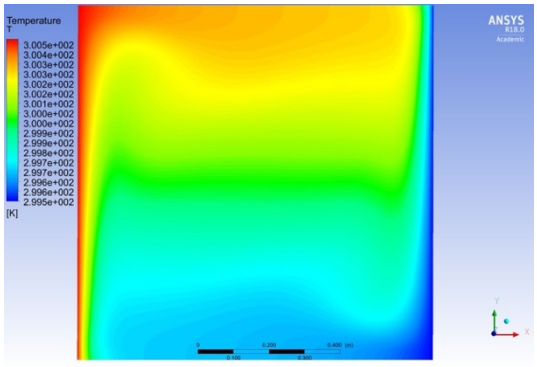
Ra = 10^3



Ra = 10^4



Ra = 10^5



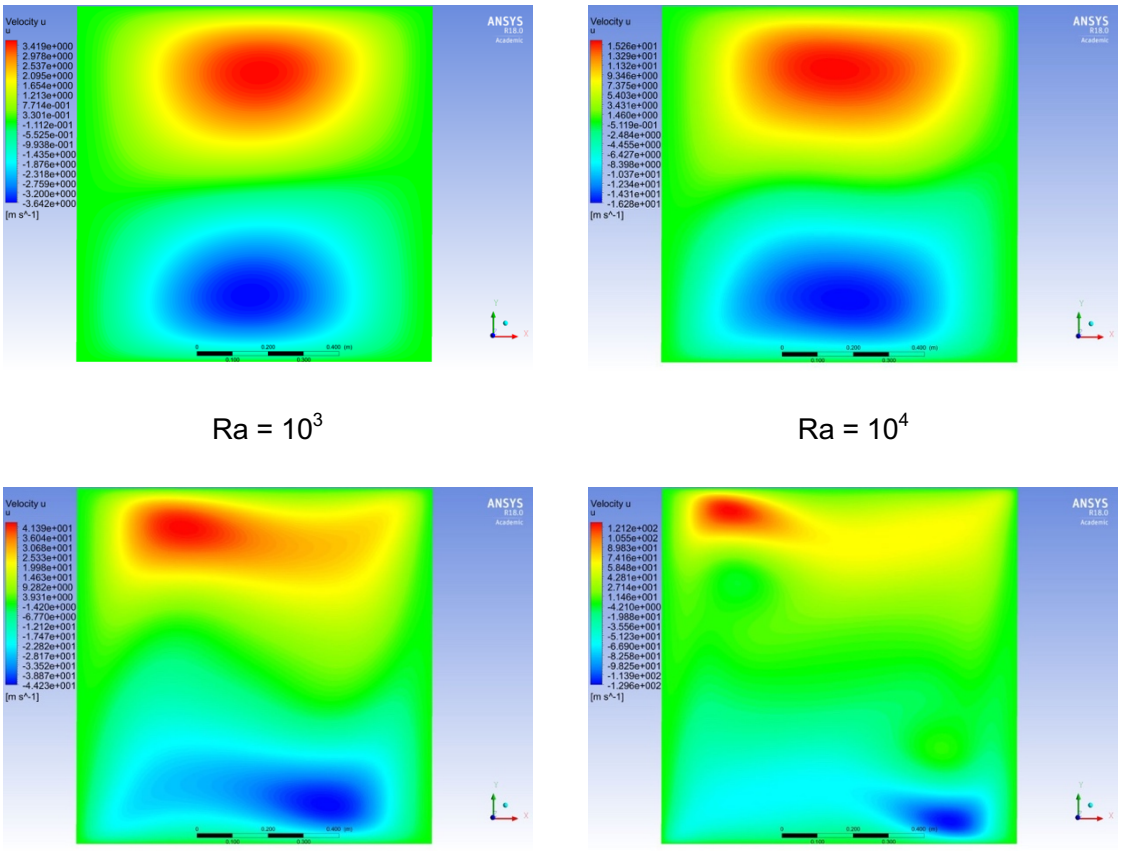
Ra = 10^6

Fig. 6.2. Temperature Isotherms for various Ra

At low Ra, the isotherms are almost vertical. This is because at this time, heat transfer is primarily in the form of conduction. As the Ra increases, the isotherms are more and more distorted to reach a horizontal arrangement. At this time, heat transfer is primarily in the form of convection.

Another parameter studied is the Nusselt number. The Nusselt number is maximum at the left bottom corner of the cavity for the hot wall and at right top corner for the cold wall. This is because there exists a maximum temperature difference in these regions and as a result, the convection heat transfer is maximum, which in turn results in the high value of Nu. The rate of heat transfer between the fluid and walls increases with the Rayleigh numbers as seen in tables of chapter 6.4.1, as the average Nu number increases along the grow of Ra.

6.4.3. Velocity distributions and streamlines

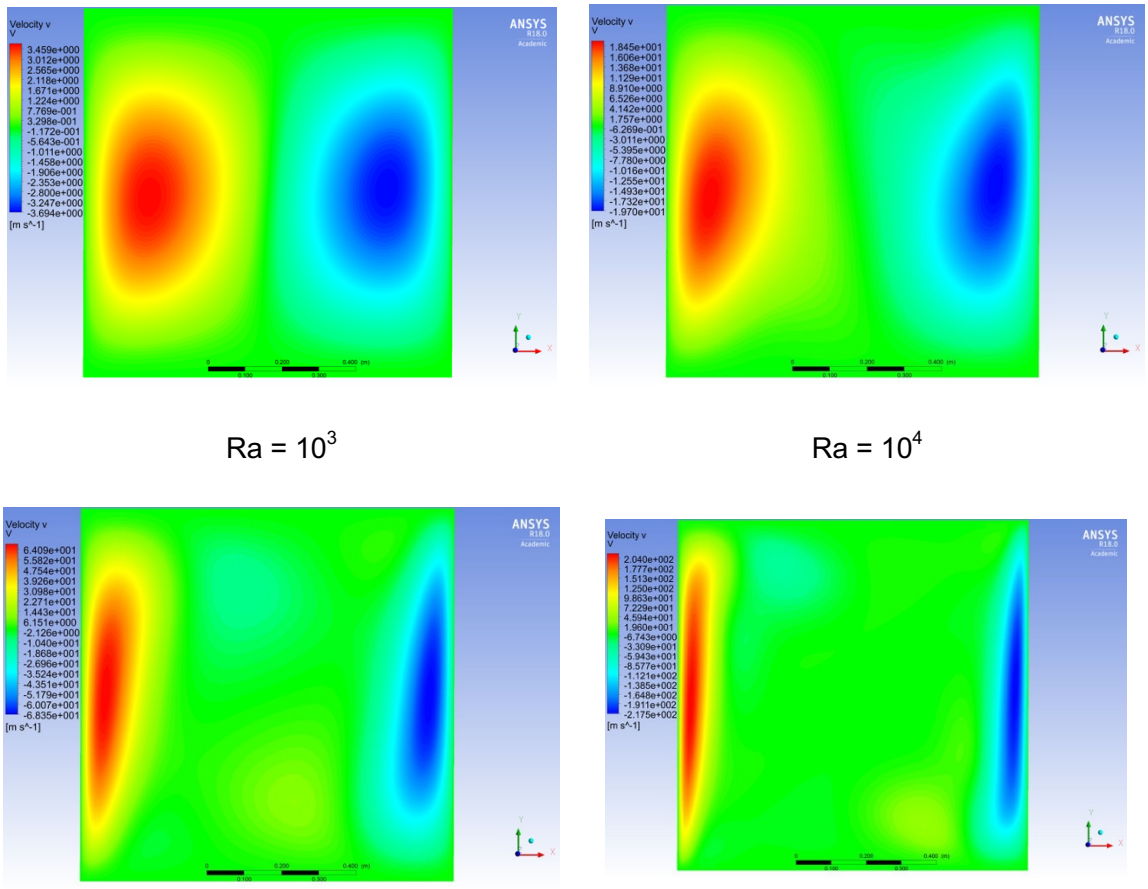


$Ra = 10^3$ $Ra = 10^4$

$Ra = 10^5$ $Ra = 10^6$

Fig. 6.3. Distribution of Horizontal Velocity for various Ra

As seen from figures above, there are two symmetrical eddies in the up and down half of the cavity, as the Ra increased, these two eddies shift and compressed towards the up and down adiabatic walls respectively.

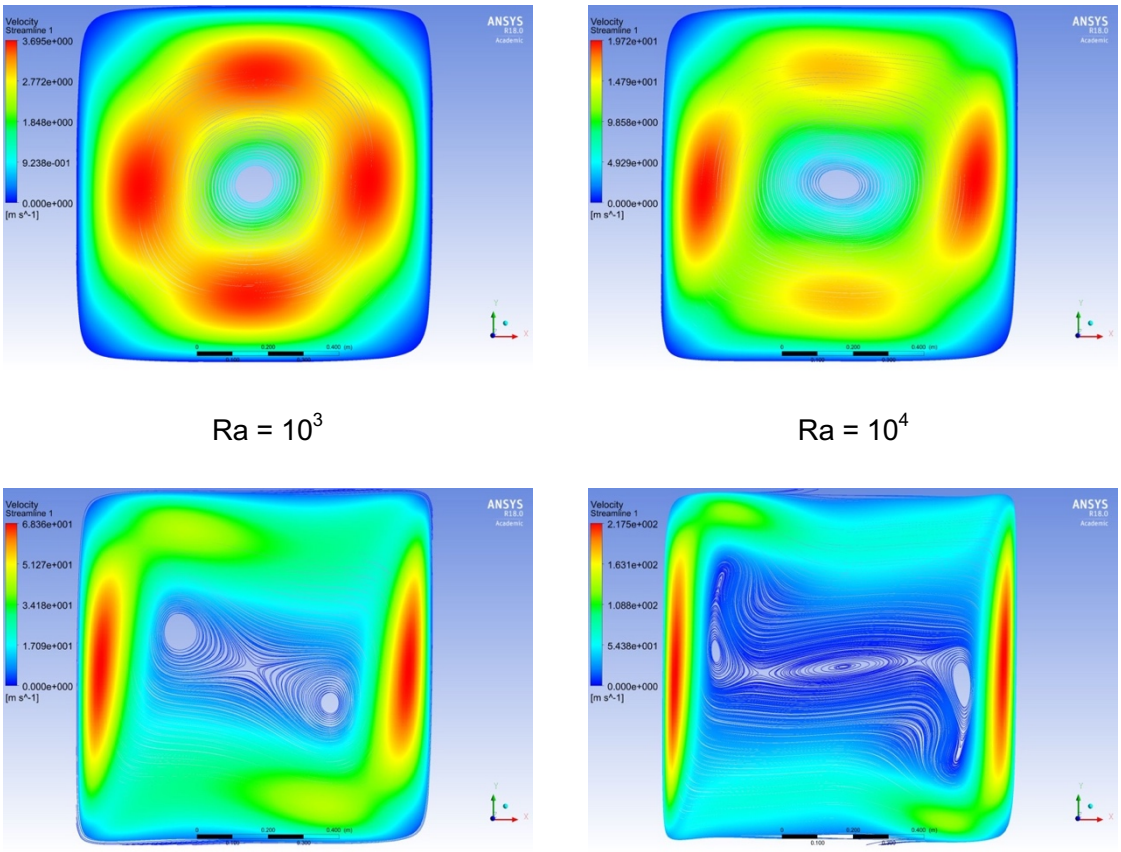


$Ra = 10^3$ $Ra = 10^4$

$Ra = 10^5$ $Ra = 10^6$

Fig. 6.4. Distribution of Vertical Velocity for various Ra

As seen from figures above, there are two symmetrical eddies in the left and right half of the cavity, as the Ra increased, these two eddies shift and compressed towards the hot and cold walls respectively.



Ra = 10^3 Ra = 10^4

Ra = 10^5 Ra = 10^6

Fig. 6.5. Streamline Contours for various Ra

The streamline contours carry an abundant amount of fluid flow characteristics on how it changes with the Rayleigh number. When the Rayleigh number is low, for Ra= 10^3 and 10^4, only s single circulating eddy exists in the central of the domain. When the Ra increases to 10^5, the central main eddy splits into two. When the Rayleigh number further increases, these two eddies tend to move towards the two corners of the cavity and there is another one formed in the center as shown.

Conclusions

In conclusion, the code developed can generate a rather precise results compared with those analytical solutions and benchmarks. This means the code developed is free of bugs, and can run to solve other problems of different cased reliably.

As the code been verified, some other conclusions which can be drawn from the project:

- When the Peclet number is high, the errors due to the first-order schemes become apparent and hence the high resolution schemes are required for obtaining a good solution
- From comparisons done for different schemes of cases studies, it is easily observed the advantage of PLS over UPS, especially with high Reynolds.
- The results using ANSYS are also in agreement with the benchmark and it offers the powerful mesh generating function, flexibility of using various discretization schemes and different solving schemes. The solutions can be obtained on comparatively coarser grids and take lesser computational time as well.

Acknowledgments

First, I would like to thank my supervisors, Prof. Ricard Consul and Prof. Rafael Ruiz for their help, guidance and continuous support throughout the project. Thanks to Ambarish Vaidya, for his work done before as my reference now. Special thanks to my friend Jian Zheng, for his support and help for my project. Thanks to professor Ivette Maria Rodriguez and Jordi Cadafalch for help me to find and chose this project.

My sincere thanks to ETSEIB, UPC, for giving me this opportunity to carry out my Master study and this final project. Last, I want to thank my parents in China for their support to my study and life here in Barcelona.

Bibliography

[1] Spalding, D. B. (1972). A novel finite difference formulation for differential expressions involving both first and second derivatives. *International Journal for Numerical Methods in Engineering*, *4*(4), 551-559.

[2] Patankar, S. V. (1981). A calculation procedure for two-dimensional elliptic situations. *Numerical heat transfer*, *4*(4), 409-425.

[3] Patankar, Suhas. *Numerical heat transfer and fluid flow*. CRC press, 1980.

[4] Smith, R. M., & Hutton, A. G. (1982). The numerical treatment of advection: A performance comparison of current methods. *Numerical Heat Transfer, Part A Applications*, *5*(4), 439-461.

[5] Harlow, F. H., & Welch, J. E. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids*, *8*(12), 2182-2189.

[6] Patankar, S. V., & Spalding, D. B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International journal of heat and mass transfer*, *15*(10), 1787-1806.

[7] Van Doormaal, J. P., & Raithby, G. D. (1984). Enhancements of the SIMPLE method for predicting incompressible fluid flows. *Numerical heat transfer*, *7*(2), 147-163.

[8] Ghia, U. K. N. G., Ghia, K. N., & Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of computational physics*, *48*(3), 387-411.

[9] Zeytounian, R. K. (2003). Joseph Boussinesq and his approximation: a contemporary view. *Comptes Rendus Mecanique*, *331*(8), 575-586.

[10] de Vahl Davis, G. (1983). Natural convection of air in a square cavity: a bench mark numerical solution. *International Journal for numerical methods in fluids*, *3*(3), 249-264.

[11] Pérez-Segarra, C. D., Consul, R., & Oliva, A. (2002). Verification of finite volume computations on steady-state fluid flow and heat transfer.

Complementary

[12] 陶文铨. *数值传热学-第 2 版*. 西安交通大学出版社, 2001.